Rockhood Automation

XC Series PLC

User Manual

Rockhood Automation Co., Ltd.

Rockhood Automation

XC series

Programmable controller

Operating manual

Version 2

This manual includes some basic precautions which you should follow to keep you safe and protect the products. These precautions are underlined with warning triangles in the manual. About other manuals that we do not mention, please follow basic electric operating rules.

Precautions

Please follow the precautions. If not, it may lead the controlsystem incorrect or abnormal, even cause fortune lose.

Correct Application

The models could only be used according to the manual, and an only be used along with the peripherial equipments recognized or recommended by Rockhood Automation. They could only work normally in the condition of be transported, kept and installed correctly, also please operate and maintain them according to the recommendation.

Duty Declare

We have checked the manual, its content fits the hardware and software of the products.As mistakes are unavoidable, we couldn’t promise all correct. However, we would check the data in the manual frequently, and in the next edition, we will correct the necessary information. Your recommendation would be highly appreciated.

Catalog

Preface

——Specialties of programmable controller

The programming of XC series programmable controller has the following characteristic:

- **Support two kinds of program languages**
  In XC series PLC，besides statement format，you can also adopt ladder chart on the screen. And，these two formats could convert to the other.
- **Rich basic functions**
  Based on the theory of "Basic functions、High speed dispose、convenient to use", XC series PLC can support not only functions relative to sequence control，but also basic application instructions of data transfer and compare、arithmetic and logic control、loop and shift of data etc.，besides, it can support interrupt、high-speed counter exclusive compare instructions、high-speed impulse output and other high-speed dispose instructions.
- **Offset function (Indirect addressing)**
  Add offset suffix after the coil、data register (e.g. X3[D100]、D0[D100]) to realize indirest addressing. E.g. when D100=0, X3[D100] means X3，D0[D100] means D0; when D100=9, X3[D100] means X14，D0[D100] means D9；
- **Single phase or AB high speed counter**
  The high speed counters in XC series PLC carry on interrupt disposal with the high speed pulse from special input points. So it is independent with the scan cycle, the count speed can reach 200KHz.
- **Convenient MODBUS communication instructions**
  With Modbus communication instruction, PLC can easily communicate with every kind of peripheral device as long as they have Modbus protocol.
- **High speed pulse output**
  The main units have two routes pulse output, output can be sequencial segments, each segment of pulse number could be set freely. The pulse could reach 400KHz.

MEMO

1．Summary of XC series PLC

XC series PLC is mini model PLC with powerful function. This series products can satisfy diverse control demand. With compact design、excellent extend capability、cheap price and powerful function, XC series PLC has become perfect resolution of small size control.

1-1. Summary of XC series PLC and program format

Introduction

XC series programmable controller

- I/O 14~60 points
- FlashROM memory inside
- Real time clock：With clock inside, Li battery power drop memory
- Multi-COM ports, can connect with inverters、instrucments、printers etc.
- Rich instructions, convenient to program

Program Format

《Statement Program》

Statement program is the format which use“LD”、“AND”、“OUT”etc. these SFC instructions to input. This format is the basic input form to compile the SFC program. But it’s not convenient for understanding.

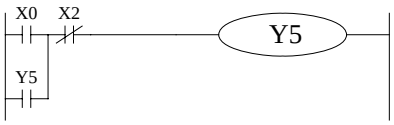
E.g：

Step	Instruction	ID
0	LD	X000
1	OR	Y005
2	ANI	X002
3	OUT	Y005

《Ladder Program》

Use sequencial control signal and soft unit’s ID to draw the sequencial circuit’s graph on the screen, which is called ladder program. As this method uses trigger point’s symbols and coil symbols to denote the sequencial control circuit，so it is easy to understand the program’s contents. At the same time it’s also available to monitor the PLC’s action via the status displayed in the circuit.

E.g：



Alternation

The programs compiled with the preceding two methods are both stored in the PLC’s program memory in the format of instruction table. So，the denotion and edition of this two program format can convert to the other.

1-2. XC series PLC's model and type

XC series Main Units

XC3—○○ □—□—□

1 2 3 4 5

1、 Serial Name
2、 I/O points
3、 Output format R：Relay output T：Transistor output
RT：Mix output of Transistor /Relay（Y0、Y1 are transistor）
4、 Supply power E：AC power C：DC power
5、 Clock S：With Clock inside

Model						Input （DC24V）	Output （R，T）
AC power			DC power				
Relay output	Transistor output	Mix output (R&T)	Relay output	Transistor output	Mix output (R&T)		
XC3-14R-E	XC3-14T-E	XC3-14RT-E	XC3-14R-C	XC3-14T-C	XC3-14RT-C	8 points	6 points
XC3-24R-E	XC3-24T-E	XC3-24RT-E	XC3-24R-C	XC3-24T-C	XC3-24RT-C	14 points	10 points
XC3-32R-E	XC3-32T-E	XC3-32RT-E	XC3-32R-C	XC3-32T-C	XC3-32RT-C	18 points	14 points
XC3-48R-E	XC3-48T-E	XC3-48RT-E	XC3-48R-C	XC3-48T-C	XC3-48RT-C	28 points	20 points
XC3-60R-E	XC3-60T-E	XC3-60RT-E	XC3-60R-C	XC3-60T-C	XC3-60RT-C	36 points	24 points

Switch quantity expansion

XC—E ○ □ ○ □

1 2 3 4 5 6

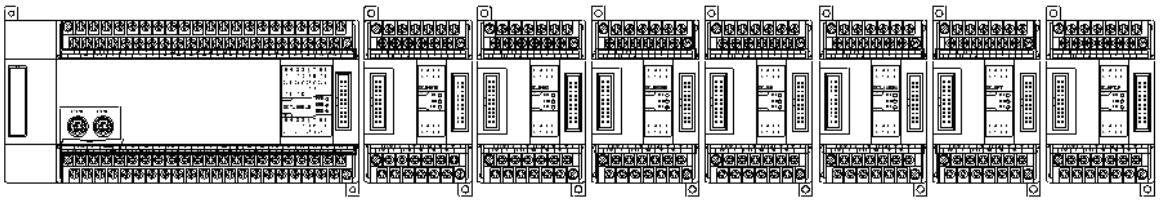
1、 Serial name
2、 E：Expansion
3、 Input points
4、 X：Input
5、 Output points
6、 Output format YR：Relay output YT：Transistor output

Model			I/O points	Input （DC24V）	Output （R，T）
Input	Relay output	Transistor output			
-	XC-E8YR	XC-E8YT	8 points	-	8 points
XC-E16X	-	-	16 points	16 points	-
-	XC-E16YR	XC-E16YT	16 points	-	16 points
-	XC-E8X8YR	XC-E8X8YT	16 points	8 points	8 points
	XC-E16X16YR	XC-E16X16YT	32 points	16 points	16 points
XC-E32X	-	-	32 points	32 points	-
-	XC-E32YR	-	32 points	-	32 points

1-3. Expansion's constitution and ID assignment

Expansion

XC series PLC can be used independently or used along with the expansions. The following is the chart of a basic unit with 7 expansions.



Constitution Rules

- Input/Output switch quantity is Octal
- Input/Output analog quantity is Decimal
- PLC main units can connect with 7 expansions and a BD module. The input/output type is not limited, both switch or analog quantity are available.

ID Assignment

Unit	Type	ID（As register）	Max points/ Channels
Expansion 1#	Input switch quantity X	X100~X137	32 points
	Output switch quantity Y	Y100~Y137	32 points
	Input analog quantity ID	ID100~ID131	16 channels
	Output analog quantity QD	QD100~QD131	16 channels
	Module’s set value D	D8250~D8259	-
Expansion 2#	Input switch quantity X	X200~X237	32 points
	Output switch quantity Y	Y200~Y237	32 points
	Input analog quantity ID	ID200~ID231	16 channels
	Output analog quantity QD	QD200~QD231	16 channels
	Module’s set value D	D8260~D8269	-
Expansion 3#	Input switch quantity X	X300~X337	32 points
	Output switch quantity Y	Y300~Y337	32 points
	Input analog quantity ID	ID300~ID331	16 channels
	Output analog quantity QD	QD300~QD331	16 channels
	Module’s set value D	D8270~D8279	-
Expansion 4#	Input switch quantity X	X400~X437	32 points
	Output switch quantity Y	Y400~Y437	32 points
	Input analog quantity ID	ID400~ID431	16 channels
	Output analog quantity QD	QD400~QD431	16 channels
	Module’s set value D	D8280~D8289	-
Expansion 5#	Input switch quantity X	X500~X537	32 points
	Output switch quantity Y	Y500~Y537	32 points
	Input analog quantity ID	ID500~ID531	16 channels
	Output analog quantity QD	QD500~QD531	16 channels
	Module’s set value D	D8290~D8299	-
Expansion 6#	Input switch quantity X	X600~X637	32 points
	Output switch quantity Y	Y600~Y637	32 points
	Input analog quantity ID	ID600~ID631	16 channels
	Output analog quantity QD	QD600~QD631	16 channels
	Module’s set value D	D8300~D8309	-
Expansion 7#	Input switch quantity X	X700~X737	32 points
	Output switch quantity Y	Y700~Y737	32 points
	Input analog quantity ID	ID700~ID731	16 channels
	Output analog quantity QD	QD700~QD731	16 channels
	Module’s set value D	D8310~D8319	-
BD Expansion	Input switch quantity X	X1000~X1037	32 points
	Output switch quantity Y	Y1000~Y1037	32 points
	Input analog quantity ID	ID1000~ID1031	16 channels
	Output analog quantity QD	QD1000~QD1031	16 channels
	Module’s set value D	D8320~D8329	-

1-4. General Specification

General Specification

Items	Specifications
Insulate voltage	Up to DC 500V 2MΩ
Anti-noise	1000V 1uS pulse per minute
Ambient temperature	0℃~60℃
Ambient humidity	5~95%
COM 1	RS-232C, connect with host machine、HMI program or debug
COM 2	RS-232C/RS-485， connect with network or aptitude instrument、inverters etc.
COM 3	CAN Bus
Installation	Can use M3 screw to fix or install directly on DIN46277（Width 35mm）orbit
Grounding	The third type grounding（can't public ground with strong power system.）

Performance

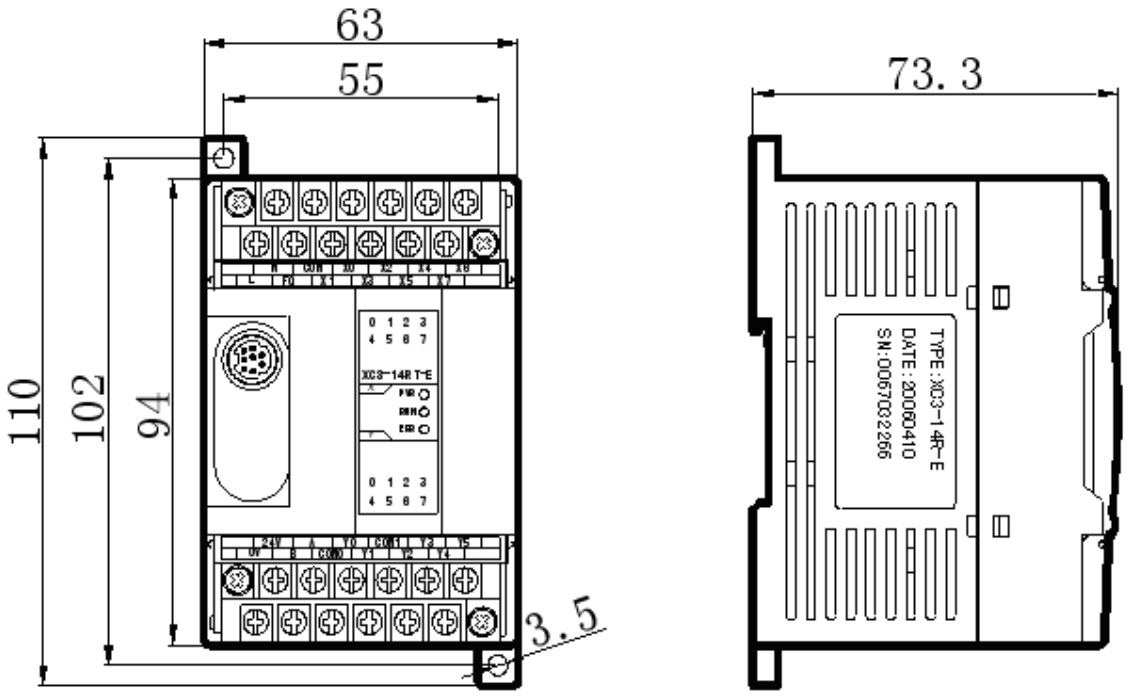
Item		Specification		
		14 points	**24\32 points**	**48\60 points**
Program executing format		Loop scan format、time scan format		
Program format		Use both statement and ladder chart		
Dispose speed		0.5us		
Power cut retentive		Use FlashROM and Li battery		
User program's capacity		30000 steps（The program is smaller than 1M）		
I/O points		8 in/ 6 output	14\18 in 10\14 out	28\36 in 20\24 out
Interior coil's points (M)		8512 points		
Timer（T）	Points No.	620 Points		
	Spec.	100mS timer：Set time 0.1~3276.7 seconds 10mS timer：Set time 0.01~327.67 seconds 1mS timer：Set time 0.01~327.67 seconds		
Counter（C）	Points No.	635 Points		
	Spec.	16 bits counter：set value K0~32767 32 bits counter：set value K0~2147483647		
Data Register（D）		8512 words		
FlashROM Register（FD）		2048 words		
High-speed counter/exterior interrupt		3 types high speed count format（Single direction、double direction、AB phase） 2 routes exterior interruption（Rising edge、Falling edge）		
Setting of time scan space		0~99mS		
Password protection		6 bits ASCII		
Self diagnose function		Power on self-diagnose、Monitor timer、grammar check		

1-5．Exterior Size

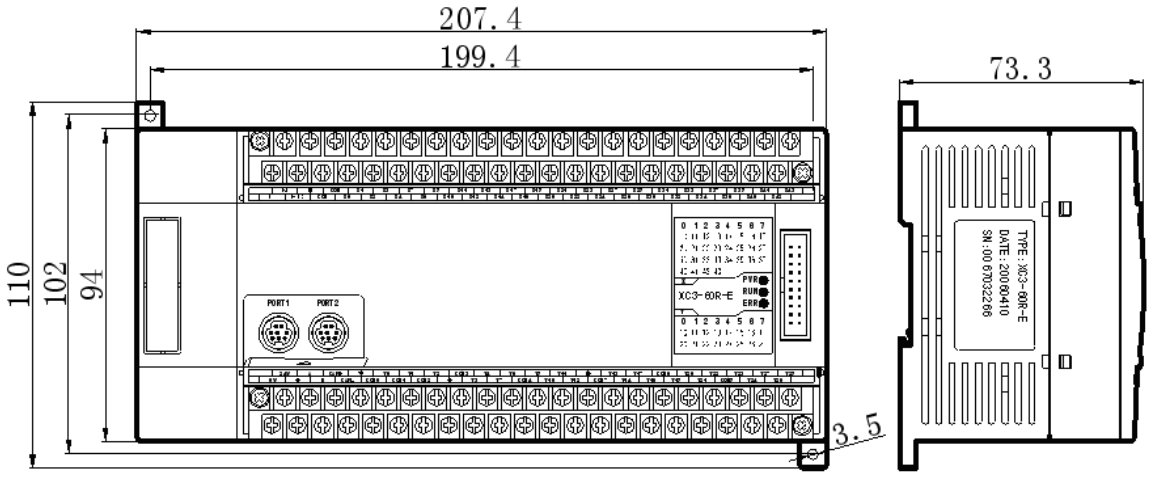
Exterior Size

XC3 series 14 points main units（Including 16 points expansions）

XC3 series 32 points main units（Including 24 points main units、32 points expansion）

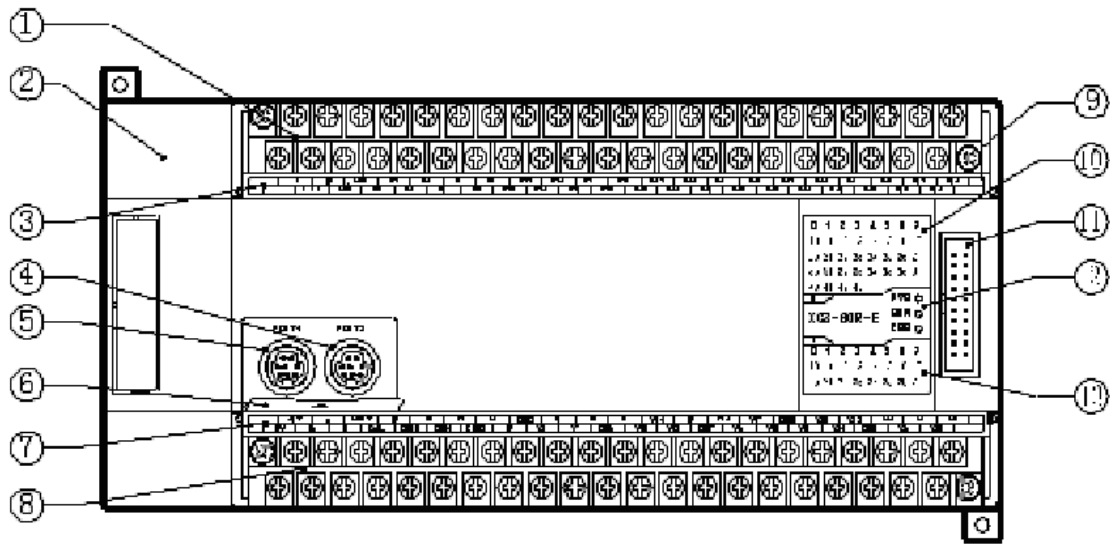


XC3 series 60 points main units（Including 48 points main units）



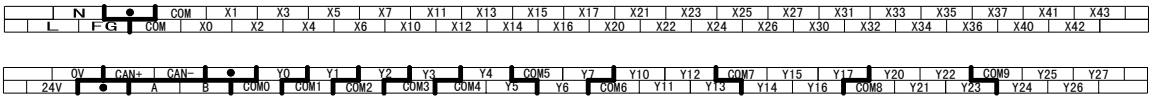
1-6. Terminal arrangement

Main Unit

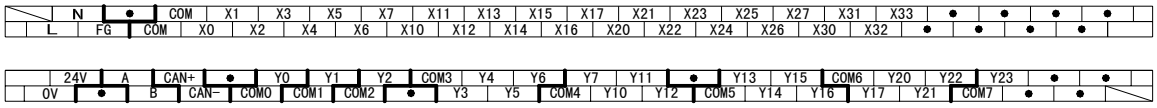


① Input terminals ② BD expansion ③ Input label ④ COM port
⑤ COM port ⑥ COM port's cover board ⑦ Output label
⑧Output terminals ⑨ Screws ⑩ Input indicate LED ⑪ Extension port
⑫Programming status indicate LED ⑬ Output indicate LED

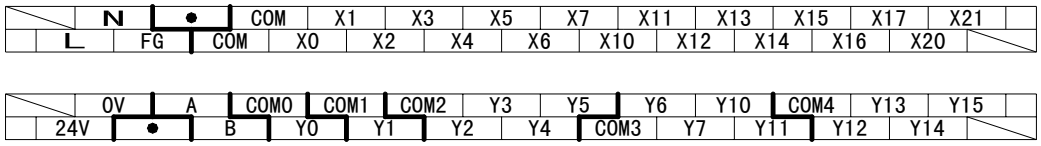
XC3- 60 main units：36 in/24 out



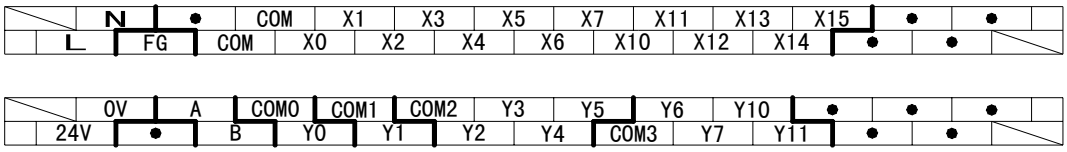
XC3- 48 main units：28 in/20 out



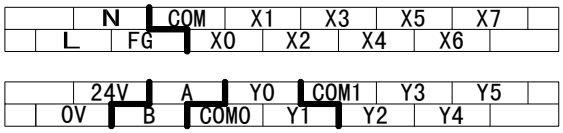
XC3- 32 main units：18 in/14 out



XC3- 24 main units：14 in/10 out

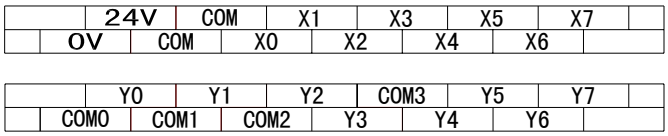


XC3- 14 main units：8 in/6 out

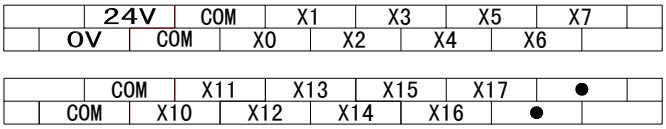


Expansion

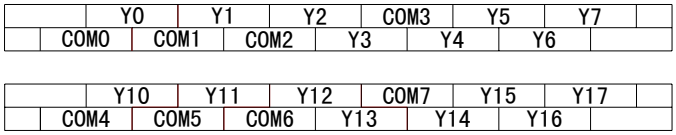
XC-E8X8YR



XC-E16X



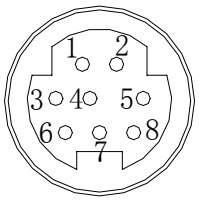
XC-E16YR



1-7. COM Port definition

COM1

Pin of COM 1：

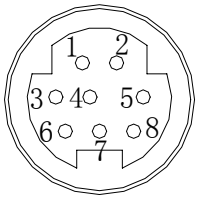


2：PRG
4：RxD
5：TxD
6：VCC
8：GND

Mini Din 8 core socket（hole）

COM2

Pin of COM 2：



4：RxD
5：TxD
8：GND

Mini Din 8 core socket（hole）

Cable

Connection of programmable cable is the following：

Mini Din 8 core socket（hole）　　DB9 pin（hole）

Memo

2. Power circuit's specification、input/output specification and exterior layout

In this chapter, we'll tell the power constitution, interior signal circuit's composing, output circuit's composing and exterior layout of XC series PLC.

When using the extend modules or special modules at the same time, please connect the power according to theuser manual.。

2-1. Power specification

For the power specification of XC series programmable controller's basic units, see the following table:

AC power type

Rated voltage	AC100V~240V
Voltage allow bound	AC90V~265V
Rated frequency	50/60Hz
Allow momentary power-cut time	Interrupt time≤0.5 AC cycle，alternation≥1 sec
Impact current	Max 40A 5mS below/AC100V max 60A 5mS below /AC200V
Max power consumption	12W
Power for sensor use	24VDC±10% max 400mA

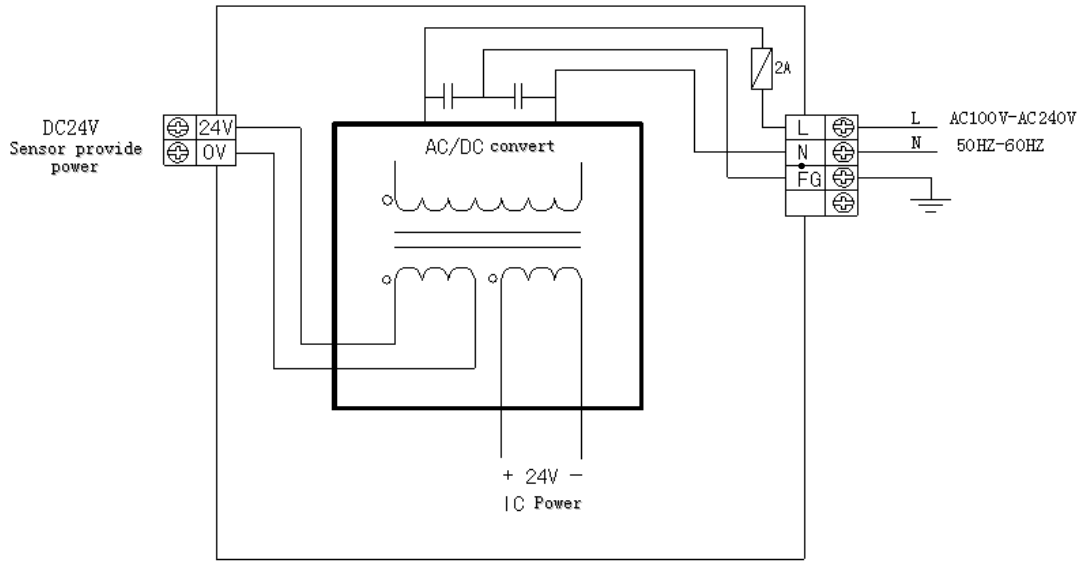
- To avoid voltage decrease, please use the power cable thicker than $2mm^2$
- Even appear power cut within 10ms，PLC can still go on working. But if long time power cut or abnormal power decrease, PLC will stop working, output will also appear OFF status，when recover power supply, the PLC will auto start to work.

DC power type

Rated voltage	DC24V
Voltage allow bound	DC21.6V~26.4V
Input current (Only basic unit)	120mA DC24V
Allow momentary power-cut time	10mS DC24V
Impact current	10A DC26.4V
Max power consumption	12W
Power for sensor use	24VDC±10% Max 400mA

2-2．AC power、DC input type

Constitution and connection

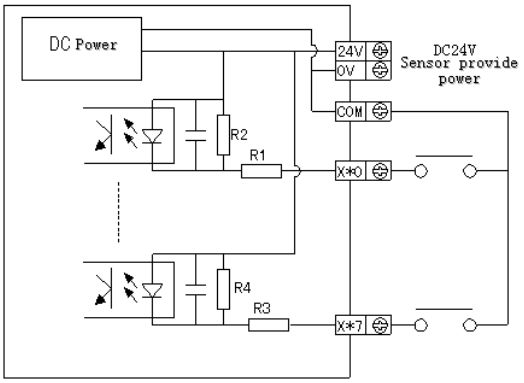


- The power is connected between L and N terminals.
- 24+、COM terminals can be used as power 400mA/DC24V which supply sensor. Besides, this terminal can't be given power from outside.
- [•] terminal is vacant terminal，please don't go on exterior connection or use it as relay terminal.
- Please connect the basic unit with extend module's [COM] terminal.

2-3. Input Specification

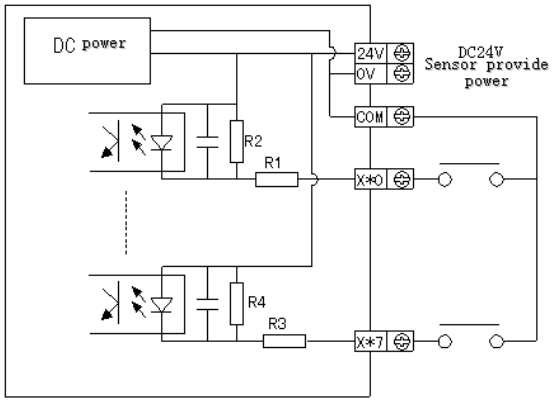
Basic units

Input signal's voltage	DC24V±10%
Input signal's current	7mA/DC24V
Input ON current	Up to 4.5mA
Input OFF current	Low than 1.5mA
Input response time	About 10ms
Input signal's format	Contact input or NPN open collector transistor
Circuit insulation	Photo-electricity coupling insulation
Input action's display	LED light when input ON



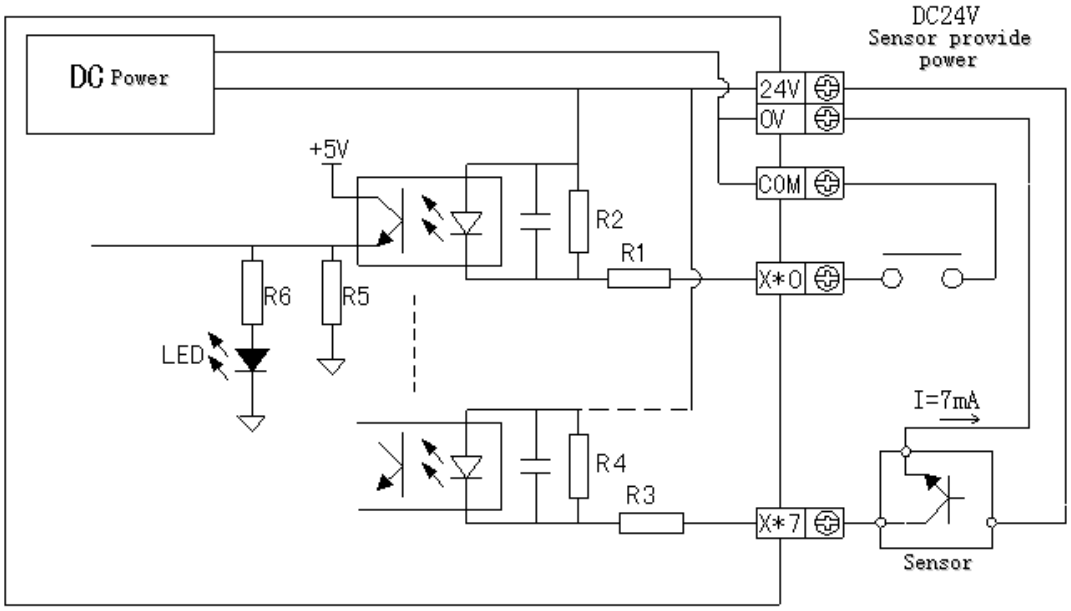
Expansions

Input signal's voltage	DC24V±10%
Input signal's current	7mA/DC24V
Input ON current	Up to 4.5mA
Input OFF current	Below 1.5mA
Input response time	About 10ms
Input signal's format	Contacts input or NPN open collector transistor
Circuit insulation	Photo-electricity coupling insulation
Input action's display	LED light when input ON.



2-4．DC input signal's disposal（AC power type）

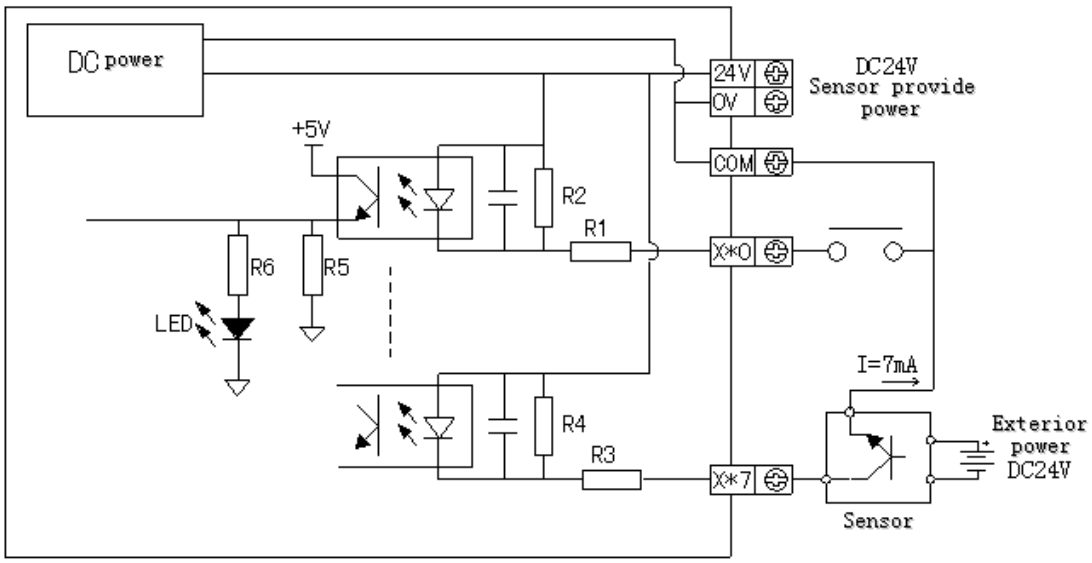
DC input signal



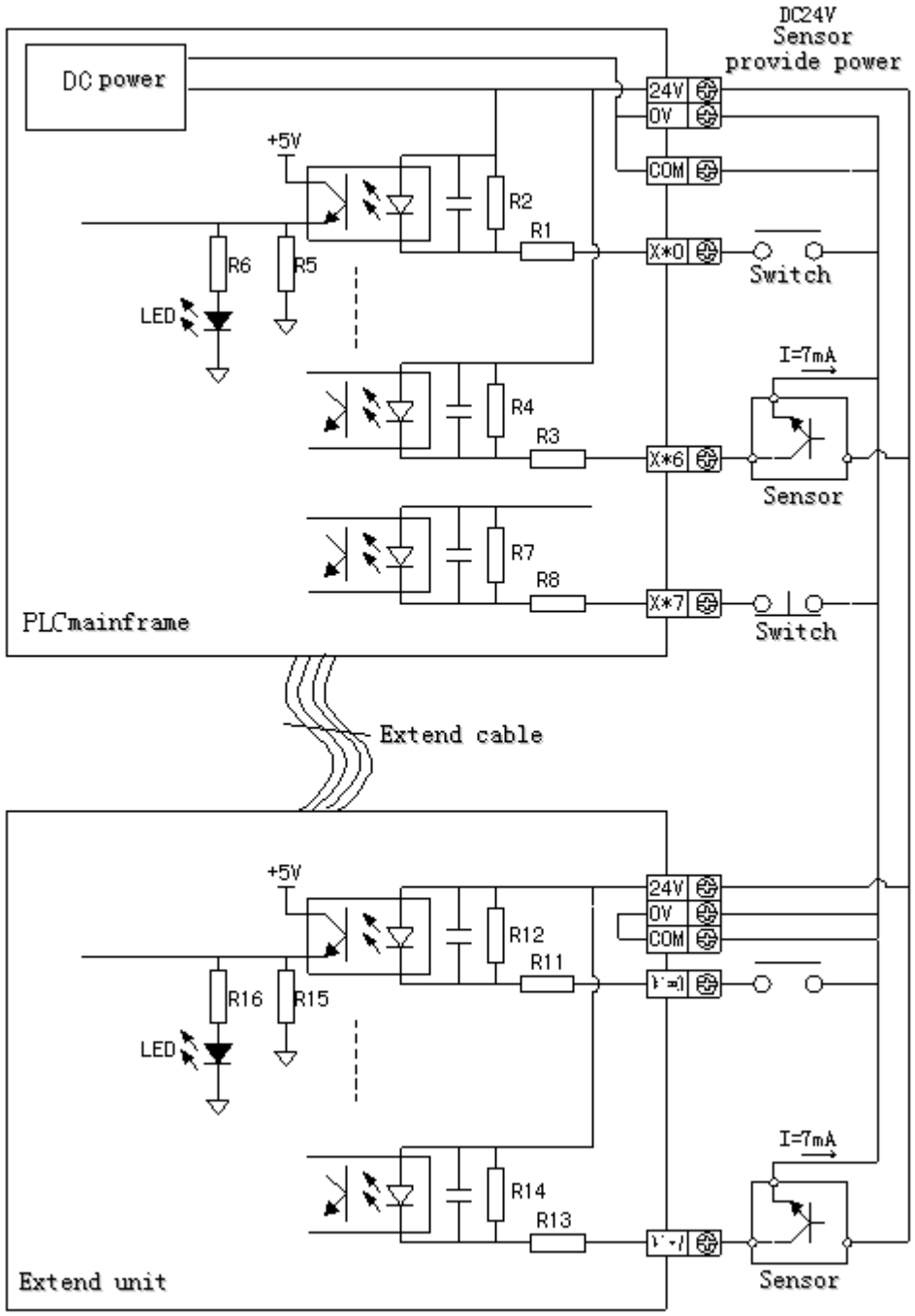
- Input terminal
  When connect input terminal and COM terminal with contacts without voltage or NPN open collector transistor, if input is ON，LED lamp lights, which indicates input。There are many COM terminals to connect in PLC.
- Input circuit
  Use optical coupling instrument to insulate the input once circuit and twice circuit，There's a C-R filter in the twice circuit。It is set to avoid wrong operation caused by vibration of input contacts or noise along with input signal. As the preceding reason, for the changing of input ON→OFF，OFF→ON，in PLC, the response time delays about 10ms。There's a digital filter inside X000~X015。This kind of filter can very from 0~15ms according to the special register (D8020).
- Input sensitive
  The PLC's input current is DC24V 7mA，but to be safe，it needs current up to 3.5mA when it's ON，lower than 1.5mA when it's OFF.

Exterior circuit used by sensor

XC series PLC's input power is supplied by its interior 24V power，so if use exterior power to drive photoelectricity sensor etc.,，this exterior power should be DC24V±4V，please use NPN open collector type for sensor's output transistor



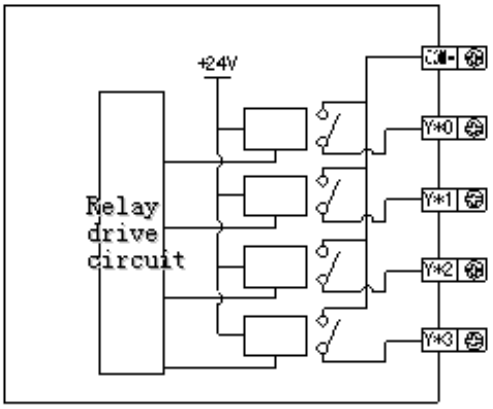
Input connection



2-5. Output specification

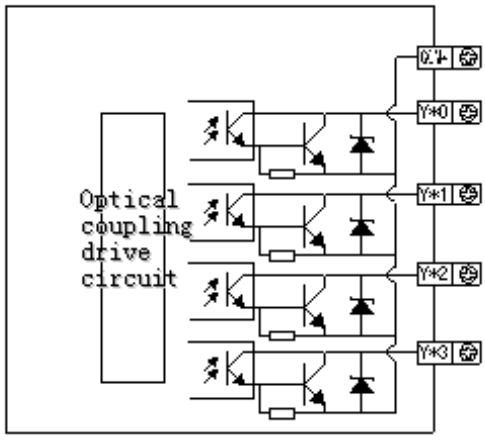
Relay output

Interior power		Below AC250V、DC30V
Circuit insulation		Mechanism insulation
Action denote		LED indicate lamp
Max load	Resistant load	3A
	Induce load	80VA
	Lamp load	100W
Open circuit's leak current		-
Mini load		DC5V 2mA
Response time	OFF→ON	10ms
	ON→OFF	10ms



Transistor Output

Interior power		Below DC5~30V
Circuit insulation		Optical coupling insulation
Action denote		Indicate lamp LED
Max load	Restance load	0.8A
	Induce load	12W/DC24V
	Lamp load	1.5W/DC24V
Open circuit's leak current		-
Mini load		DC5V 2mA
Response time	OFF→ON	Below 0.2ms
	ON→OFF	Below 0.2ms

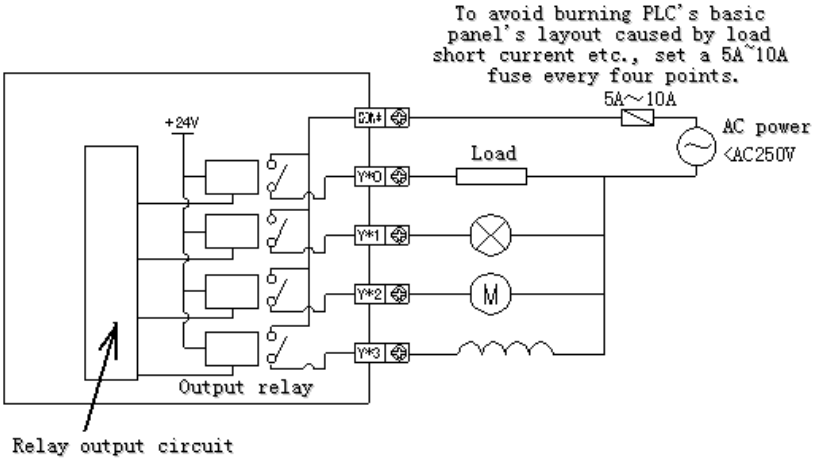


2-6．Disposal of relay output circuit

Relay output circuit

- **Output terminals**
  Relay output type includes 2~4 public terminals. So each public-end unit can drive different power-voltage system's (E.g.：AC200V，AC100V，DC24V etc.) load.
- **Circuit's insulation**
  Between the relay output coils and contacts，PLC's interior circuits and exterior circuits, load circuits are electric insulation. Besides, each public-end blocks are separate.
- **Action display**
  LED lamp lights when output relay's coils galvanize, output contacts are ON.
- **Response time**
  From the output relay galvanize (or cut) to the output contacts be ON (or OFF), the response time is about 10ms
- **Output current**
  The current-voltage below AC250V can drive the load of pure resistace 2A/1 point、inductance load below 80VA（AC100V or AC200V） and lamp load below 100W（AC100V or AC200V）.
- **Open circuit's leak current**
  When the output contact be OFF and there's no leak current，can directly drive Ne lamp etc.
- **The life of relay output contacts**
  Standard life of induce AC load such as contactor、electromagnetism valve：5 million times for 20VAload。Cut power device's life according to the company's test: for 80VA load, the action life is up to 2 million times.

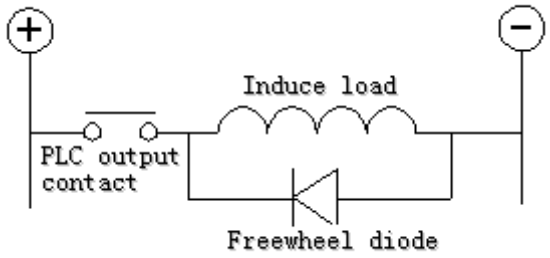
Output connection example



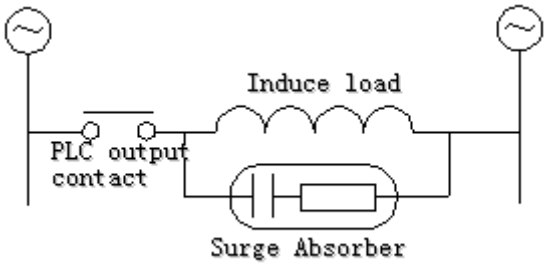
Constitution of output circuit

- For DC induce load, please parallel connect with commutate diode. If not connect with the commutate diode, the contact's life will be decreased greatly. Please choose the commutate diode which allow inverse voltage endurance up to 5~10 times of the load's voltage, ordinal current exceeds load current.
- Parallel connect AC induce load with surge absorber can reduce noise.

DC load



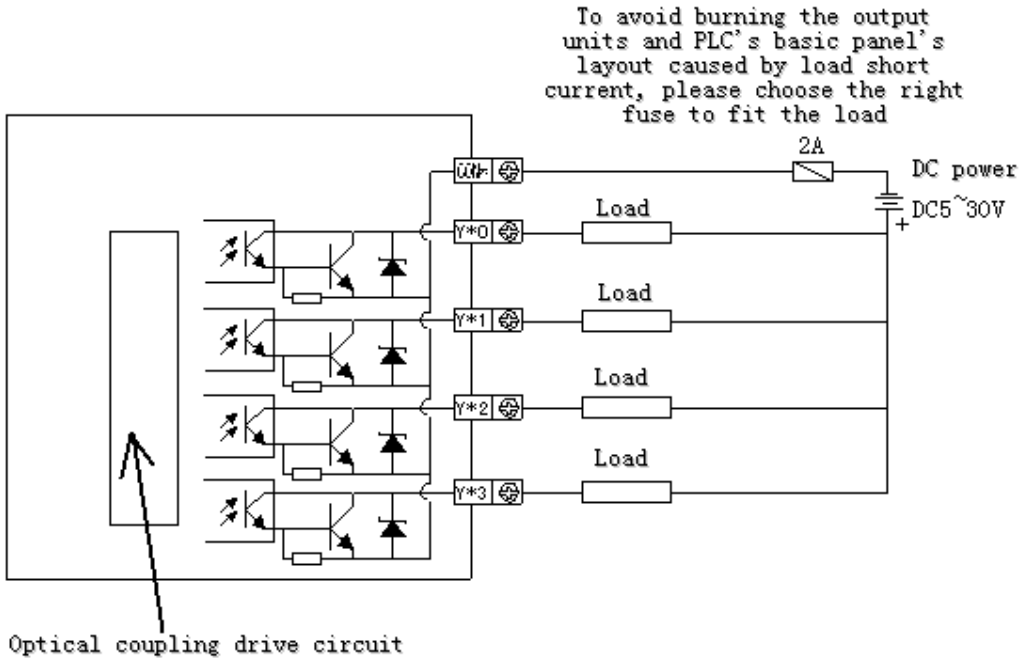
AC load



2-7. Disposal of transistor output circuit

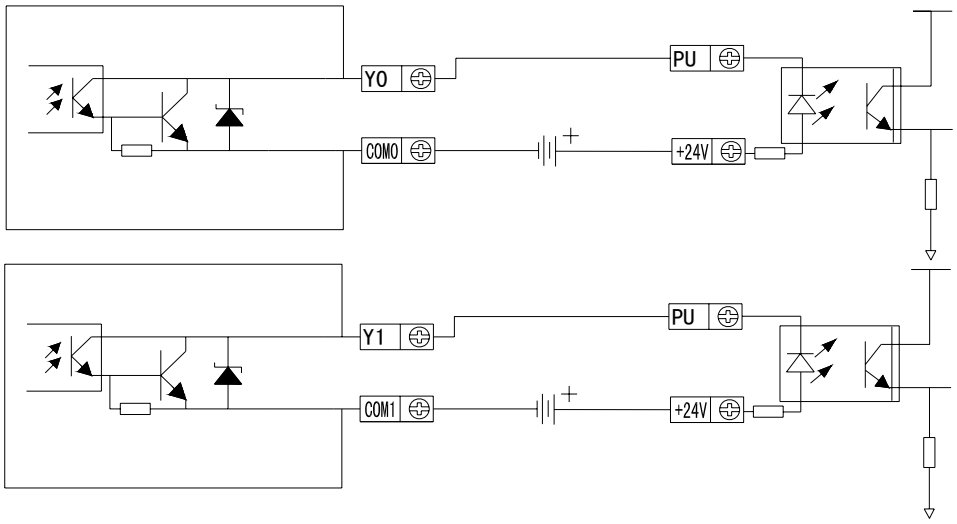
Transistor output circuit

- Output terminal
  Basic unit's transistor output has 1~4 public-end output.
- Exterior power
  Please use DC5~30V steady-voltage power for load drive,
- Circuit insulation
  Use photoelectricity coupling device to insulate PLC's interior circuit and output transistor. Besides, each public block is separate.
- Action denote
  When drive optical coupling，LED lights，output transistor is ON.
- Response time
  From photoelectricity coupling device drive (or cut) to transistor ON (or OFF), the time PLC uses is below 0.2ms.
- Output current
  The current is 0.5A per point。But as restrict by temperature goes up, the current is 0.8A every four points.
- Open circuit's current
  Below 0.1mA



E.g. ： The following is the connection graph of –RT type PLC with step motor driver。

PLC side　　Step motor driver



MEMO

3. Each soft unit's usage and function

This chapter, we'll give some description of the PLC's data and the function of interior input/output relay, auxiliary relay, status, counter, data register etc. This item is the base to use PLC.

3-1. Every soft unit of programmable controller

In the programmable controller, there are many relays、timers and counters，they all have countless "a" contacts（Normally open contacts）and "b" contacts（Normally closed contacts），Connect these contacts and coils to constitute sequencial control circuit. The following, we'll briefly introduce each soft unit

【Input（X）and output（Y）relay】

- In each basic unit，assign the ID of input relay, output relay in the format of X000~X007，X010~X017…,Y000~Y007，Y010~Y017… this octal format. The ID of extension is connected behind basic unit.
- The ID of expansion obeys the principle of channel 1 starts from X100/Y100, channel 2 starts from X200/Y200… 7 expansions could be connected totally.
- Use digital filter in the special input filter of input relay, so you can use the program to change the sieve value. So in the high-speed receive application, you can assign this type of relay's ID No.

【Auxiliary relay（M）】

- Auxiliary relay is the relay inside the programmable controller，this type of output relay is different from input/output relay，it can't gain exterior input，it also can't drive exterior load，it can only be used in the program。
- The relay used for retentive can still save its ON/OFF status in the case of PLC power cut.

【Status（S）】

- Relay used as step ladder chart.
- When not used as working procedure No.，it's the same with auxiliary relay and can be used as common contact/coil to carry on programming. Besides, it can also be signal alarm to diagnose exterior trouble.

【Timer（T）】

- Timer could carry on plus operation to 1ms，10ms，100ms etc. time pulse in PLC, When reach certain set value, output contact act.
- T100~T199 are timers with the unit of 100ms clock pulse，their current values are the accumulate values. So, even though timer coil's drive input is cut, they will still hold the current value, go on accumulating the action.

【Counter（C）】

- The counters can be divided into the following sorts according the their usage and purpose：

  [Used for interior count] Common use / power failure retentive use

  16 bits counter：Used for plus count，count bound: 1~32,767

  32 bits counter：Used for add / minus count，count bound: -2,147,483,648~+2，147,483,647

  These counters are used for PLC's interior signals, usually their response speed is below 10Hz.

  [Used for high-speed count] For power failure retentive use

  32 bits counter: For plus / minus count, count bound: -2,147,483,648~+2,147,483,6487 (Single phase plus count，single phase plus/minus count，AB phase count) allocate to the special input points.

  High-speed counter can count with the frequency below 200kHz，independent with the PLC's scan cycle.

【Data register（D）】

- Data register is the soft unit used by data register to save data. XC series PLC's data registers are all 16 bits (The high bit is the sign bit), Combine two registers can carry on 32 bits data disposal (The high bit is the sign bit).

  Just the same with other soft units, data registers can also be divided to be two types: for common use and power failure retentive use.

【Constant（K）、(H)】

- In the diverse value used by PLC, K means decimal integer，H means Hex. Value. They are used to be the set value and current value for the timer and counter, or applied instructions' operands.

【Pointer（P）（I）】

- Pointers are used for branch and interrupt. The pointer (P) used by branch is the jump aim used for condition jump or subroutine jump. Pointer used for interrupt is used for the assigned input interrupt, time interrupt.

3-2. Device's ID list

For the allocate of device's ID, please see the following list:
Besides, when connect input / output expansions and special expansions on the basic units, for the input / output relay's No., please refer to the user manual.

Mnemonic	Name	Bound			points		
		14 points	24\32 points	48 \60 points	14 points	24\32 points	48 \60 points
X	Input relay	X000~X007	X000~X015 X000~X021	X000~X033 X000~X043	8 points	14\18 points	28\36 points
Y	Output relay	Y000~Y005	Y000~Y011 Y000~Y015	Y000~Y023 Y000~Y027	6 points	10\14 points	20\24 points
M	Interior relay	M0~M2999【M3000~M7999】			8000		
		M8000~M8511 for special using			512		
S	Flow	S0~S511 【S512~S1023】			1024		
T	Timer	T0~T99：100ms not accumulation			620		
		T100~T199：100ms accumulation					
		T200~T299：10ms not accumulation					
		T300~T399：10ms accumulation					
		T400~T499：1ms not accumulation					
		T500~T599：1ms accumulation					
		T600~T618：1ms with interruption precise time					
C	Counter	C0~C299：16 bits forth counter			635		
		C300~C598：32 bits forth/back counter					
		C600~C634：high-speed counter					
D	Data Register	D0~D3999 【D4000~D7999】			8000		
		For special usage D8000~D8511			512		
FD	FlashROM Register	FD0~FD1535			1536		
		For special usage FD8000~FD8511			512		

◆ **NOTE：**

※1. The memorizer area in 【 】 is the defaulted power failure retentive area; soft elements D、M、S、T、C can be set to change the power failure retentive area. For the details, please see the following table

※2. FlashROM register needn't set power failure retentive, its data won't lose when power is cut (No battery).

※3. The serial No. of input coil、output relay are octal data, other memorizers' No. are all algorism data.

Setting of soft unit's power failure saving area

Mnemonic	Set area	Function	System's defaulted value	Memory bound of power drop
D	FD8202	Start denotation of D power cut save area	4000	D4000~D8000
M	FD8203	Start denotation of M power cut save area	3000	M3000~M8000
T	FD8204	Start denotation of M power cut save area	620	Not set
C	FD8205	Start denotation of C power cut save area	320	C320~C640
S	FD8206	Start denotation of S power cut save area	512	S512~S1024

3-3．Data disposal of programmable controller

According to different usage and purpose, XC series programmable controllers use 5 types of count format. For their usage and function, see the following：

《DEC》（DEC：DECIMAL　NUMBER）

- The set value of timer and counter（K constant）
- The ID of auxiliary relay（M），timer（T），counter（C），status（S）（Soft unit's number）
- Assign the value in the operands and instruction's action（K constant）

《HEX》（HEX：HEXADECIMAL　NUMBER）

- The same with DEC data，it is used to assign the value in the operands and instruction's action（H constant）

《BIN》（BIN：BINARY　NUMBER）

- Just as said before，carry on data allocation to timer, counter or data register in the format of DEC. or Hex., But in the PLC, these data are all be put in the format of binary data. And, when carry on monitor on the peripherial device, these soft units will auto switch to be DEC. data as shown in the graph.（they can also switch to be Hex. Data.）.

《OCT》（OCT：OCTAL　NUMBER）

- The input relay, output relay's soft units' ID of XC series PLC are allocate in the format of OCT data. So, it can go on carry of [1-7，10-17，. . . 70-77，100-107].

《BCD code》（BCD：　BINARY　CODE　DECIMAL）

- BCD is the method which use 4 bits binary to denote decimal 0~9. It's easy to despose bit. So, BCD is available to denote digital switch or 7 segments display control.

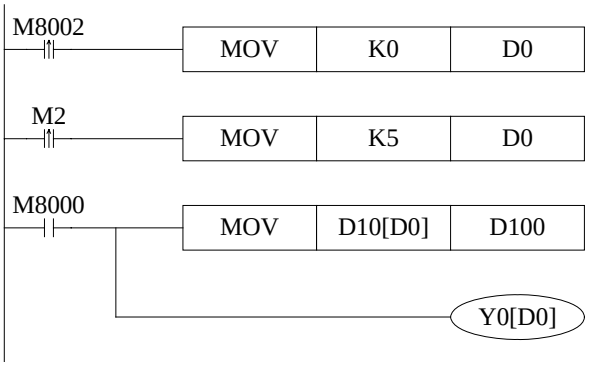
《Other data（float）》

- XC series PLC has the function of high precision floating point operation. Use binary floating point data to execute floating point operation, use decimal floating value to execute monitor.

3-4. Some encode principles of device

1、Data register could be used as offset（indirect assignment）

Format：Dn[Dm]、Xn[Dm] 、Yn[Dm] 、Mn[Dm] etc.



In the preceding example, when D0=0, then D100=D10, Y0 is ON；

When M2 turns from OFF to be ON, D0=5, then D100=D15, Y5 is ON.

When D10[D0]=D[10+D0], Y0[D0]=Y[0+D0].

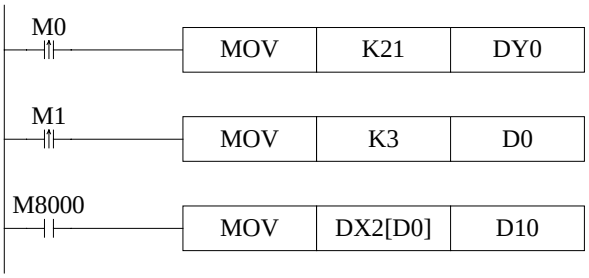
- Word's offset composed by bit soft units：DXn[Dm] means DX[n+Dm]；
- Soft units with offset, the offset could only be denoted with soft device D.

2、 Bit units compose word

Input X、output Y、middle coil M could compose 16 bits word. E.g. DX0 means X0~X17compose to be a 16 bits data. DX20 means X20~X37 combines a 16 bits data.

Format：Add a D before bit device

Bit devices combine to be word devices：DX、DY、DM、DS、DT、DC



In the preceding example, when M0 turns from OFF to be ON, the value of the word DY0 composed by Y0~Y17 equals 21，i.e. Y0、Y2、Y4 turns to be ON status.

Before M1 be activate, when D0=0, DX2[D0] means a word composed by X2~X21；

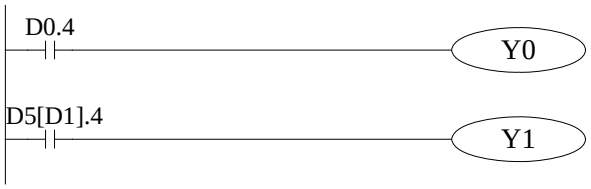
When M1turns from OFF to be ON, D0=3, then DX2[D0] means a word composed by X5~X24

- DXn（the bound of "n" is the exact bound of "X"），choose 16 points from the head to the end, add 0 if not enough.
- Please note, the word composed by bit device couldn't carry on bit searching address.

3、 Bit of word device

Format：Dn.m

Register could carry on bit searching address, e.g. Dn.m means number "m" bit of Dn data register（0≤m≤15）。



In the preceding example, D0.4 means when the No.4 bit of D0 is 1, Y0 set ON；

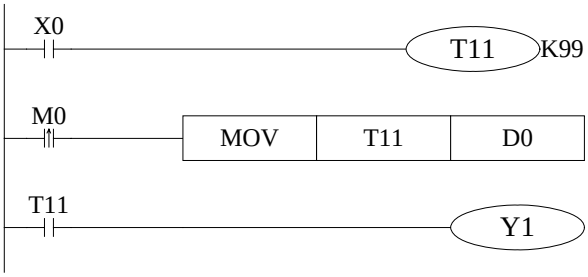
D5[D1].4 means bit searching address with offset, if D1=5，it says D5[D1] means the number 4 bit of D10.

- The bit of word device with offset is denoted as Dn[Dm].x
- Please note, to the bit of word device, they couldn't combined to be word device.

4、 T/C means the difference of register's word and bit

To T and C register, Tn/Cn means be a bit register or a word register should be distinguished by the instructions.

T、C could denote the status of timer、counter, or the current value of time、counter, it is distinguished by the instructions.



In the preceding example，MOV T11 D0，T11 means word register；

LD T11，T11 means bit register.

5、 Tag type：P，I

e.g.: P means the tag which using CJ instruction or CALL instruction which could jump; I means interrupt tag.

3-5. Timer's number and function [T]

Timer's number

Please see the following table for the timer's [T] number（the number is assigned according to Hex.）

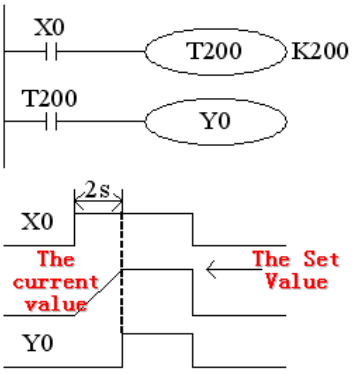
100ms not accumulated（16 bits）	T0~T99
100ms accumulated（16 bits）	T100~T199
10ms not accumulated（16 bits）	T200~T299
10ms accumulated（16 bits）	T300~T399
1ms not accumulated（16 bits）	T400~T499
1ms accumulated（16 bits）	T500~T599
1ms with interrupt precise time（32 bits）	T600~T618 (T600,T602....T618) (each engrosses 2 timers' number) the number should be even

Function

The timer accumulates clock pulse of 1ms，10ms，10ms inside PLC. When reach the set value, the output contact activates.

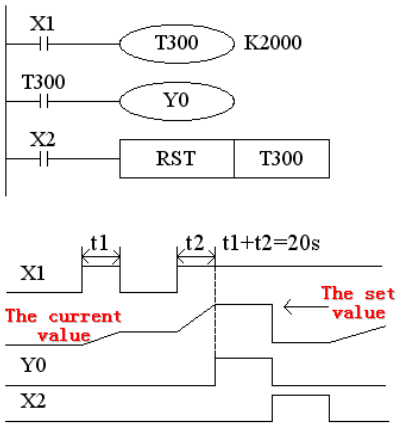
The common timers don't set exclusive instructions, use OUT instruction to time; use constant K in the program memory, also you could use register's content (D) to indirect assign.

Common format



If drive input X000 of time coil T200 is ON, T200 accumulates 10ms clock pulse with the current value timer. If this current value equals the set value K200, timer's output contact activates. That is, output contact activates after 2 seconds of coil driving. Driving input X000 cut or power cut, timer reset, output contact reset.

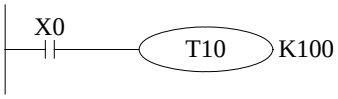
Accumulation format



If the drive input X001 of timer's coil T300 is ON, T300 accumulates 10ms clock pulse with the current value counter. When the value reaches the set value K2000, counter's output contact activates. In the count process, even the input X001 cut or drop power, when start again, go on counting, its accumulation time is 20 seconds. When reset input X002 is ON, timer reset, output contact reset.

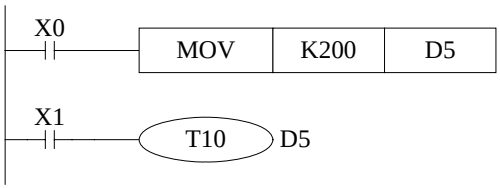
Assign method of the set value

《Constant assignment (K)》



T10 is a timer with the unit of 100ms. Assign 100 as a constant, then 0.1s × 100=10s timer work.。

《Indirect assignment (D)》

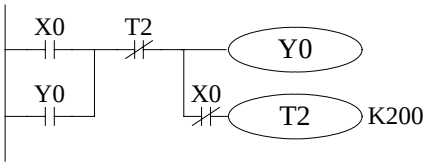


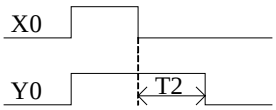
Write content in indirect data register to program or input via data switch.

When assigned as power cut retentive register, please note that voltage low will cause the set value instable.

Action

《Output delay on-off timer》

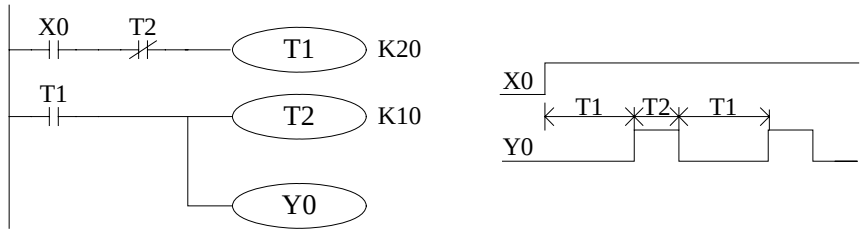




When X000 is ON, output Y000;

If X000 changes from ON to be OFF, T2（20 seconds）will be delayed, then will output Y000 cut.

《Flicker》



3-6. Counter's ID and function [C]

Counter's ID

For the counter's number（C）, please see the following table.

16 bits positive counter	C0~C299
32 bits positive/negative counter	C300~C598 (C300,C302...C598) (Each one engrosses 2 counter No.) The number must be even.
High speed counter	C600~C634(C600,C602...C634) (Each one engrosses 2 counter No.) The number must be even

Counter's characteristic

The characters of 16 bits counter and 32 bits counter are the following.

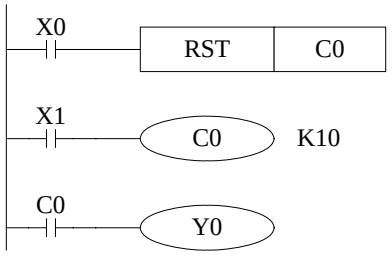
Items	16 bits counter	32 bits counter
Count direction	Positive	Positive/negative
The set value	1~32,767	-2,147,483,648~+2,147,483,647
The assigned set value	Constant K or data register	Same as the left, but data register must be in a couple
Changing of the current value	Change after positive count	Change after positive count (Loop counter)
Output contact	Hold the action after positive count	Hold the action after positive count, reset if negative count
Reset activates	When executing RST command，counter's current value is 0，output contacts recover	
The current value register	16 bits	32 bits

Function

About the assignment of normally used counter and power failure retentive counter could be changed in the method of changing FD parameters' setting via the peripheral device.

16 bits counter For normally use or power count retentive

16 bits binary increase counter, its valid setting value is K1~K32,767 (Decimal constant). The set value K0 and K1 have the same meaning, i.e. act when output contacts at the beginning of first time count.

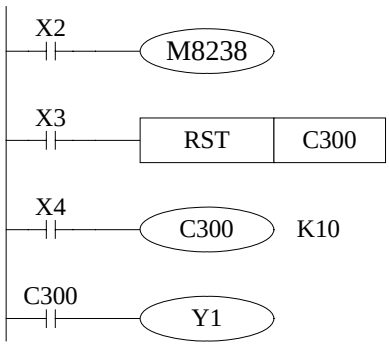


If cut the PLC's power, then the value of the normally use counter will be cleared. However, counter used by power cut retentive could save the count value after power cut, and the counter will go on counting from the value.

- Every time when X001 drives coil C0, the counter's current value will increase. When execute the coil instruction the tenth time, output contact acts. Later, even X001 activates, counter's current value will not change.
- If reset input X000 is ON, execute RST instruction, counter's current value is 0, output contacts activates.
- For the counter's set value, it could not only set by constant K, but also be assigned by data register's ID. E.g. assign D10, if the content of D10 is 123, it's the same with setting K123.
- When write the set value to the current value register via MOV instruction etc. When input next time, output coil gets, current value register turns to the set value.

32 bits counter For normally use or power count retentive

For 32 bits binary increase counter, its valid bound is K1~K2,147,483,647 (Decimal constant). With special auxiliary relay M8238, assign the direction of bits positive/negative counter's (C300~C498) direction



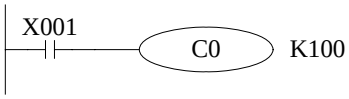
- If X2 drives M8238, then it is negative count; If no drive, then it is positive count.
- According to constant K or to the content of data register D, set the value to be positive. Treat contents in consecutive data register as a pair, and dispose it as 32 bits data. So, when assign D0, dispose D0 and D1 as a 32 bits set data. If use count input X004 to drive coil C300, execute increase count.

- When reset input X3 is ON, execute RST instruction, counter's current value turns to be 0, output contact resets.
- When use counter as power cut retentive, counter's current value, output contact's action and reset status cut power retentive.
- 32 bits counter can also be used as 32 bits data register. But 32 bits data register can't be used as device in 16 bits applied instructions.

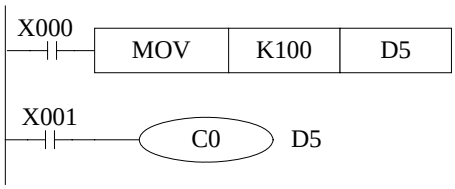
Assign method of the set value

◆ **16 bits counter**

《Constant assignment（K）》



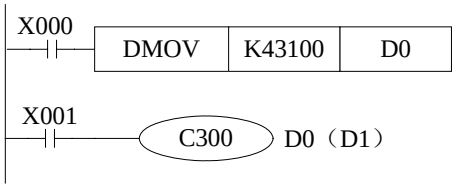
《Indicate assignment（K）》



◆ **32 bits counter**

《Constant assignment（K）》

《Indicate assignment（K）》



3-7. Some points to note

《Action order of input/output relay and response delay》

- Input disposal

  Before PLC executing the program, read all the input terminal's ON/OFF status of PLC to the image area. In the process of executing the program, even the input changed, the content in the input image area will not change. However, in the input disposal of next scan cycle, read out the change.

- Output disposal

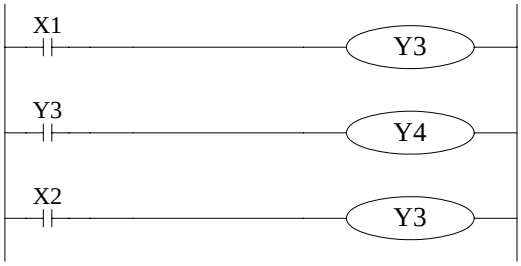
  Once finish executing all the instructions, transfer the ON/OFF status of output Y image area to the output lock memory area. This will be the actual output of the PLC. The contacts used for the PLC's exterior output will act according to the device's response delay time.

  When use this input/output format in a batch, the drive time and operation cycle of input filter and output device will also appear response delay.

《Not accept narrow input pulse signal》

PLC's input ON/OFF time should be longer than its loop time. If consider input filter's response delay 10ms, loop time is 10ms，then ON/OFF time needs 20 ms separately. So, up to 1，000/（20+20）=25Hz input pulse can't be disposed. But, this condition could be improved when use PLC's special function and applied instructions.

《Dual output（Dual coils）action》



When executing dual output (use dual coil),the back side act in prior

As showed in the left map, please consider the things of using the same coil Y003 at many position:
E.g. X001=ON，X002=OFF

At first, X001 is ON, its image area is ON, output Y004 is also ON.

But, as input X002 is OFF, the image area of Y003 is OFF.

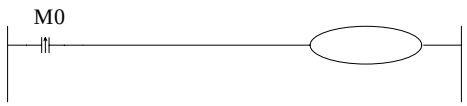
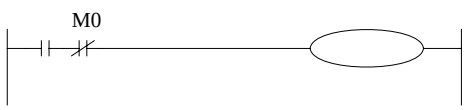
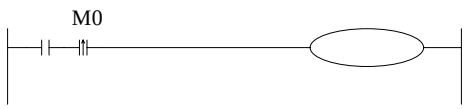
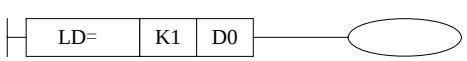
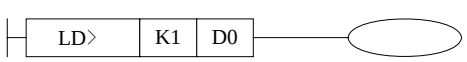
So, the actual output is : Y003=OFF,
Y004= ON.

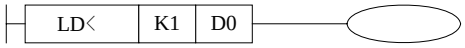
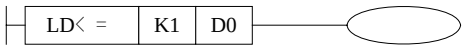
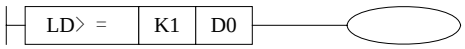
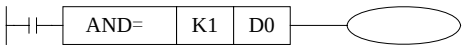
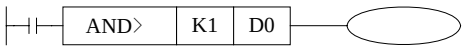
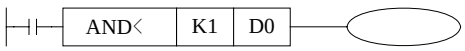
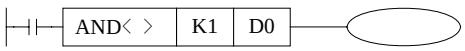
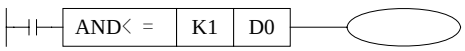
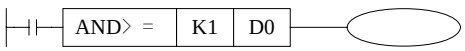
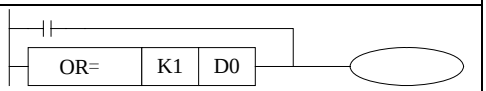
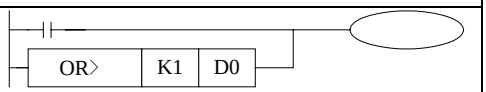
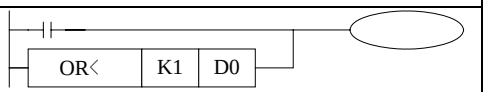
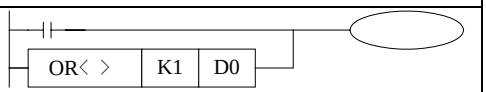
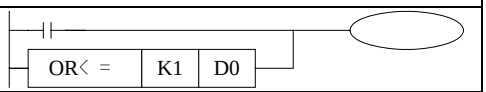
Memo

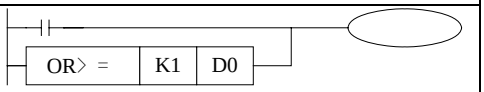
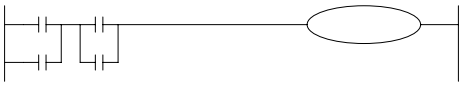
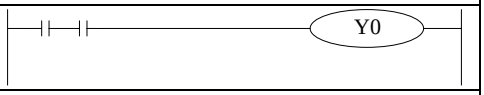
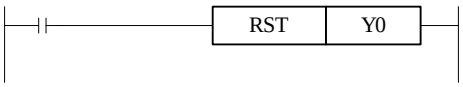
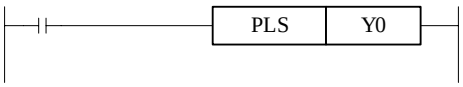
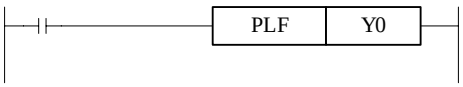
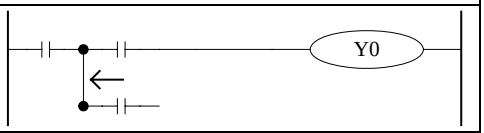
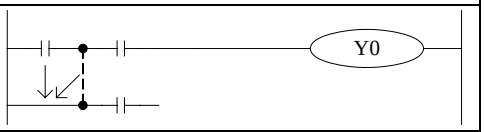
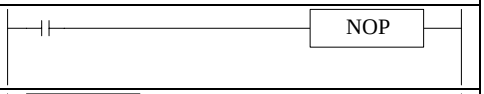
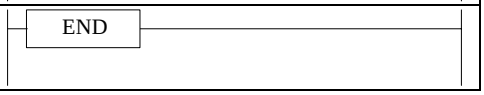
4. Basic program instructions

In this chapter, we tell some basic instructions and their functions.

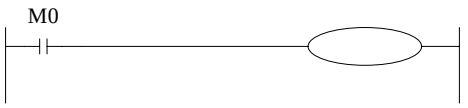
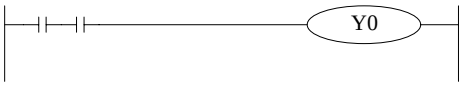
4-1. List of basic instructions

Mnemonic	Function	Format and device
LD (LoaD)	Initial logical operation contact type NO (normally open)	
LDI (LoaD Inverse)	Initial logical operation contact type NC (normally closed)	
LDP (LoaD Pulse)	Initial logical operation-Rising edge pulse	
LDF (LoaD Falling Pulse)	Initial logical operation-Falling /trailing edge pulse	
AND (AND)	Serial connection of NO (normally open) contacts	
ANI (AND Inverse)	Serial connection of NC (normally closed) contacts	
ANDP (AND Pulse)	Serial connection of rising edge pulse	
ANDF (AND Falling pulse)	Serial connection of falling/trailing edge pulse	
OR (OR)	Parallel connection of NO (normally open) contacts	
ORI (OR Inverse)	Parallel connection of NC (normally closed) contacts	
ORP (OR Pulse)	Parallel connection of rising edge pulse	
ORF (OR Falling pulse)	Parallel connection of falling/trailing edge pulse	
LD＝	Initial comparison contact. Active when the comparison （S1）＝（S2） is true.	
LD＞	Initial comparison contact. Active when the comparison （S1）>（S2） is true	

LD<	Initial comparison contact. Active when the comparison （S1）<（S2） is true	
LD<>	Initial comparison contact. Active when the comparison （S1）≠（S2） is true	
LD<=	Initial comparison contact. Active when the comparison （S1）≤（S2） is true	
LD>=	Initial comparison contact. Active when the comparison （S1）≥（S2） is true	
AND=	Serial comparison contact. Active when the comparison （S1）=（S2） is true.	
AND>	Serial comparison contact. Active when the comparison （S1）>（S2） is true.	
AND<	Serial comparison contact. Active when the comparison （S1）<（S2） is true.	
AND<>	Serial comparison contact. Active when the comparison （S1）≠（S2） is true.	
AND<=	Serial comparison contact. Active when the comparison （S1）≤（S2） is true.	
AND>=	Serial comparison contact. Active when the comparison （S1）≥（S2） is true.	
OR=	Parallel comparison contact. Active when the comparison （S1）=（S2） is true.	
OR>	Parallel comparison contact. Active when the comparison （S1）>（S2）is true.	
OR<	Parallel comparison contact. Active when the comparison （S1）<（S2）is true.	
OR<>	Parallel comparison contact. Active when the comparison （S1）≠（S2）is true.	
OR<=	Parallel comparison contact. Active when the comparison	

	（S1）≤（S2）is true.	
OR＞＝	Parallel comparison contact. Active when the comparison （S1）≥（S2）is true.	
ANB (ANd Block)	Serial connection of multiply parallel circuits	
ORB (OR Block)	Parallel connection of multiply parallel circuits	
OUT (OUT)	Final logic operation type coil drive	
SET (SET)	Set a bit device permanently ON	
RST (ReSeT)	Reset a bit device permanently OFF	
PLS (PuLSe)	Rising edge pulse	
PLF (PuLse Falling)	Falling/trailing edge pulse	
MCS (New bus line start)	Connect the public serial contacts	
MCR (Bus line return)	Clear the public serial contacts	
ALT (Alternate state)	The status of the assigned device is inverted on every operation of the instruction	
NOP (No Operation)	No operation or null step	
END (END)	Force the current program scan to end	

4-2.【LD】,【LDI】,【OUT】

Mnemonic	Function	Format and device X,Y,M,S,T,C
LD (LoaD)	Initial logic operation contact type NO (Normally Open)	
LDI (LoaD Inverse)	Initial logic operation contact type NC (Normally Closed)	
OUT (OUT)	Final logic operation type drive coil	

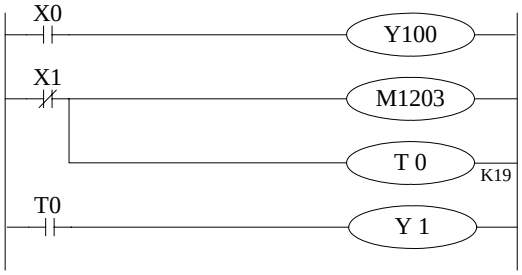
Instruction description

- Connect the LD and LDI instructions directly to the left bus bar. Or use them to define a new block of program when using ANB instruction.
- OUT instruction is the coil drive instruction for the output relay、auxiliary relay、status、timer、counter. For the input relay, cannot use.
- Can not sequentially use parallel OUT command for many times.
- For the timer's time coil or counter's count coil, after using OUT instruction, set constant K is necessary.
- For the constant K's set bound、actual timer constant、program's step relative to OUT instruction（include the set value）

See the following table

Timer/counter	Setting bound of K	The actual set value
1ms timer	1～32,767	0.001～32.767 seconds
10ms timer		0.01～32.767 seconds
100ms timer		0.1～32.767 seconds
16 bits counter	1～32,767	Same as the left
32 bits counter	1～＋2,147,483,647	Same as the left

Program




```
LD      X0
OUT     Y100
LDI     X1
OUT     M1203
OUT     T0
SP      K19
LD      T0
OUT     Y1
```

4-3.【AND】,【ANI】

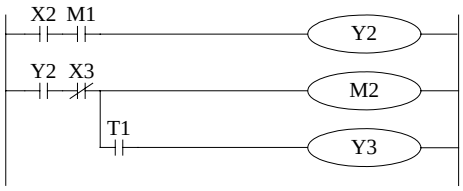
Mnemonic and Function

Mnemonic	Function	Format and device X,Y,M,S,T,C
AND (AND)	Serial connection of NO (Normally Open) contacts	
ANI (ANd Inverse)	Serial connection of NC (Normally Closed) contacts	

Description

- Use the AND and ANI instructions for serial connection of contacts. As many contacts as required can be connected in series. They can be used for many times.
- The output processing to a coil, through writing the initial OUT instruction is called a “follow-on” output (For an example see the program below: OUT M10 and OUT Y005). Follow-on outputs are permitted repeatedly as long as the output order is correct. There’s no limit for the serial connected contacts’ No. and follow-on outputs’ number.

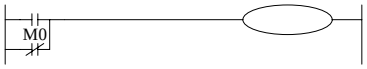
Program




```
LD    X2
AND   M1
OUT   Y2
LD    Y2
ANI   X3
OUT   M2
AND   T1
OUT   Y3
```

4-4.【OR】,【ORI】

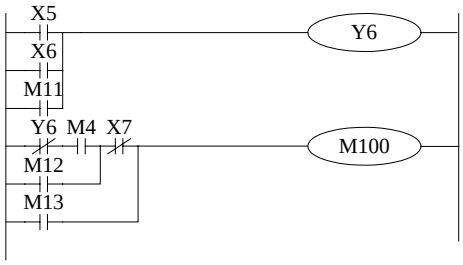
Mnemonic and Function

Mnemonic	Function	Format and device X,Y,M,S,T,C
OR (OR)	Parallel connection of NO (Normally Open) contacts	
ORI (OR Inverse)	Parallel connection of NC (Normally Closed) contacts	

Description

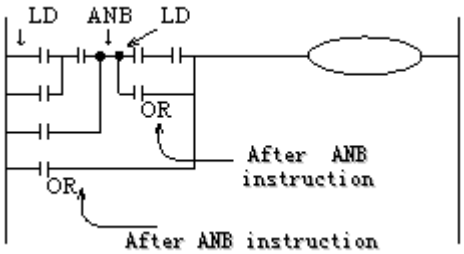
- Use the OR and ORI instructions for parallel connection of contacts. To connect a block that contains more than one contact connected in series to another circuit block in parallel, use an ORB instruction.
- OR and ORI start from the instruction's step, parallel connect with the LD and LDI instruction's step said before. There is no limit for the parallel connect times.

Program




```
LD    X5
OR    X6
OR    M11
OUT   Y6
LDI   Y6
AND   M4
OR    M12
ANI   X7
OR    M13
OUT   M100
```

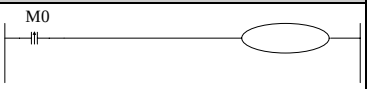
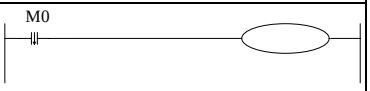
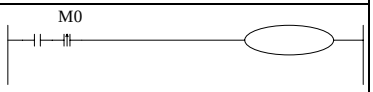
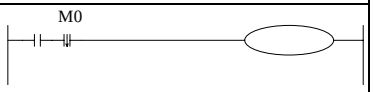
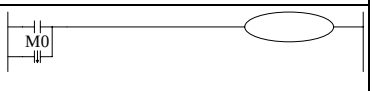
Relationship with ANB



The parallel connection with OR,ORI instructions should connect with LD, LDI instructions in principle. But after the ANB instruction, it's available to add a LD or LDI instruction.

4-5.【DP】,【LDF】,【ANDP】,【ANDF】,【ORP】,【ORF】

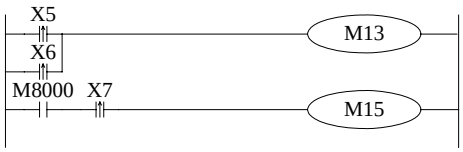
Mnemonic and Function

Mnemonic	Function	Format and device X,Y,M,S,T,C
LDP (LoaD Pulse)	Initial logical operation-Rising edge pulse	
LDF (LoaD Falling pulse)	Initial logical operation Falling/trailing edge pulse	
ANDP (AND Pulse)	Serial connection of Rising edge pulse	
ANDF (AND Falling pulse)	Serial connection of Falling/trailing edge pulse	
ORP (OR Pulse)	Parallel connection of Rising edge pulse	
ORF (OR Falling pulse)	Parallel connection of Falling/trailing edge pulse	

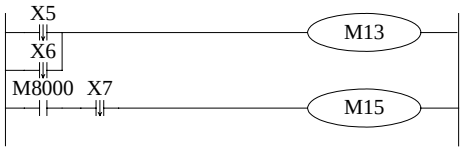
Description

- LDP、ANDP、ORP are active for one program scan after the associated device switches from OFF to ON.
- LDF、ANDF、ORF are active for one program scan after the associated device switches from ON to OFF.

Program



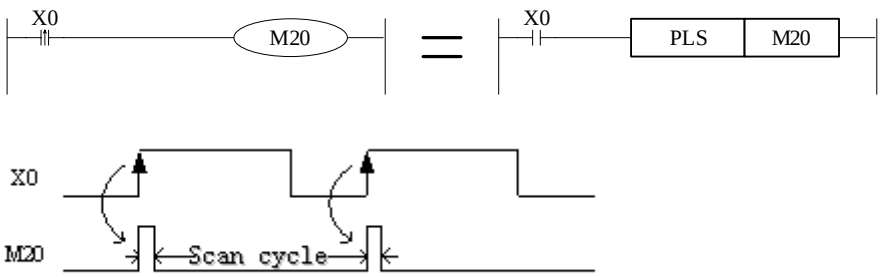

```
LDP     X5
ORP     X6
OUT     M13
LD      M8000
ANDP    X7
OUT     M15
```




```
LDF     X5
ORF     X6
OUT     M13
LD      M8000
ANDF    X7
OUT     M15
```

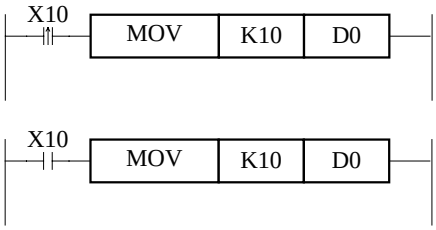
In the preceding chart, when X005～X007 turns from ON to OFF or from OFF to ON, M13 or M15 has only one scan cycle activates.

Output drive



In two conditions, when X0 turns from OFF to ON, M20 gets a scan cycle.

NOTE:



When X10 turns from OFF to ON, only execute once MOV instruction.

When X10 turns from OFF to ON, each scan cycle execute once MOV instruction.

4-6. Contacts compare instruction

Mnemonic and Function

Mnemonic	Function
LD＝	Initial comparison contact. Active when the comparison （S1）＝（S2） is true.
LD＞	Initial comparison contact. Active when the comparison （S1）>（S2） is true
LD＜	Initial comparison contact. Active when the comparison （S1）<（S2） is true
LD＜＞	Initial comparison contact. Active when the comparison （S1）≠（S2） is true
LD＜＝	Initial comparison contact. Active when the comparison （S1）≤（S2） is true
LD＞＝	Initial comparison contact. Active when the comparison （S1）≥（S2） is true
AND＝	Serial comparison contact. Active when the comparison （S1）＝（S2） is true.
AND＞	Serial comparison contact. Active when the comparison （S1）>（S2） is true.
AND＜	Serial comparison contact. Active when the comparison （S1）<（S2） is true.
AND＜＞	Serial comparison contact. Active when the comparison （S1）≠（S2） is true
AND＜＝	Serial comparison contact. Active when the comparison （S1）≤（S2） is true.
AND＞＝	Serial comparison contact. Active when the comparison （S1）≥（S2） is true.
OR＝	Parallel comparison contact. Active when the comparison （S1）＝（S2） is true.
OR＞	Parallel comparison contact. Active when the comparison （S1）＞（S2）is true.
OR＜	Parallel comparison contact. Active when the comparison （S1）＜（S2）is true.
OR＜＞	Parallel comparison contact. Active when the comparison （S1）≠（S2）is true.
OR＜＝	Parallel comparison contact. Active when the comparison （S1）≤（S2）is true.
OR＞＝	Parallel comparison contact. Active when the comparison （S1）≥（S2）is true.

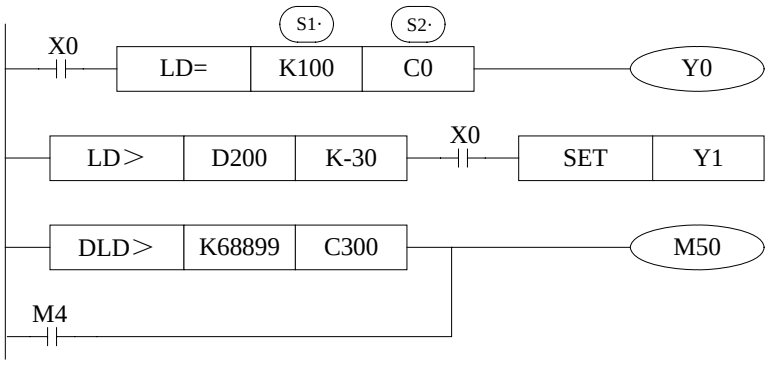
LD □

Format and Function

The value of S1 and S2 are tested according to the comparison of the instruction. If the comparison is true then the LD contact is active. If the comparison is false then the LD contact is not active.

16 bits	32 bits	Active condition	Inactive condition
LD＝	DLD＝	（S1）＝（S2）	（S1）≠（S2）
LD＞	DLD＞	（S1）＞（S2）	（S1）≤（S2）
LD＜	DLD＜	（S1）＜（S2）	（S1）≥（S2）
LD＜＞	DLD＜＞	（S1）≠（S2）	（S1）＝（S2）
LD＜＝	DLD＜＝	（S1）≤（S2）	（S1）＞（S2）
LD＞＝	DLD＞＝	（S1）≥（S2）	（S1）＜（S2）

Program



Note Items

- When the source data's highest bit (16 bits：b15，32 bits：b31) is 1，use the data as a negative.
- The comparison of 32 bits counter（C200~） must use 32 bits instruction. If assigned as 16 bits instruction, it will lead the program error or operation error.

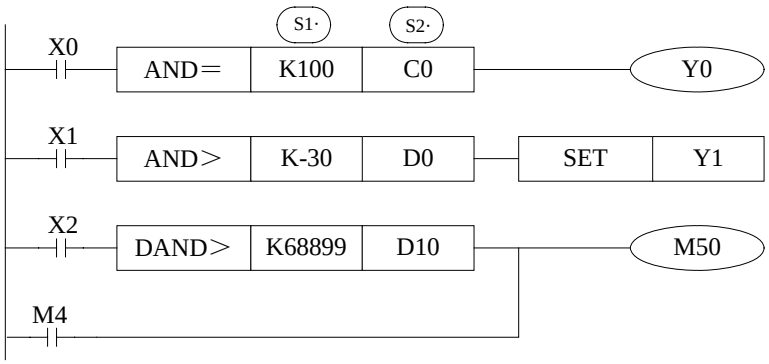
AND □

Format and Function

The value of S1 and S2 are tested according to the instruction. If the comparison is true then the AND contact is active. If the comparison is false then the AND contact is not active.

16 bits	32 bits	Active condition	Inactive condition
AND=	DAND=	（S1）＝（S2）	（S1）≠（S2）
AND>	DAND>	（S1）＞（S2）	（S1）≤（S2）
AND<	DAND<	（S1）＜（S2）	（S1）≥（S2）
AND<>	DAND<>	（S1）≠（S2）	（S1）＝（S2）
AND<=	DAND<=	（S1）≤（S2）	（S1）＞（S2）
AND>=	DAND>=	（S1）≥（S2）	（S1）＜（S2）

Program



Note items

- When the source data's highest bit (16 bits：b15，32 bits：b31) is 1，use the data as a negative.
- The comparison of 32 bits counter（C200~） must be 32 bits instruction. If assigned as a 16 bits instruction, it will lead the program error or operation error.

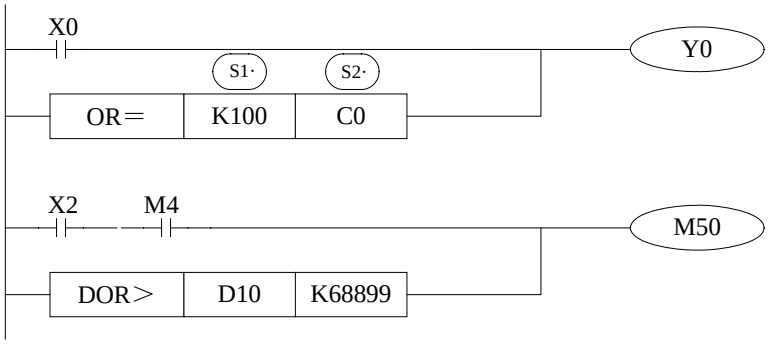
OR □

Format and Function

The value of S1 and S2 are tested according to the instruction. If the comparison is true then the OR contact is active. If the comparison is false then the OR contact is not active.

16 bits	32 bits	Active condition	Inactive condition
OR＝	DOR＝	（S1）＝（S2）	（S1）≠（S2）
OR＞	DOR＞	（S1）＞（S2）	（S1）≤（S2）
OR＜	DOR＜	（S1）＜（S2）	（S1）≥（S2）
OR＜＞	DOR＜＞	（S1）≠（S2）	（S1）＝（S2）
OR＜＝	DOR＜＝	（S1）≤（S2）	（S1）＞（S2）
OR＞＝	DOR＞＝	（S1）≥（S2）	（S1）＜（S2）

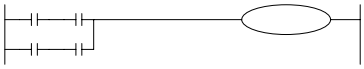
Program



Note items

- When the source data's highest bit (16 bits：b15，32 bits：b31) is 1，use the data as a negative.
- The comparison of 32 bits counter（C300~） must be 32 bits instruction. If assigned as a 16 bits instruction, it will lead the program error or operation error.

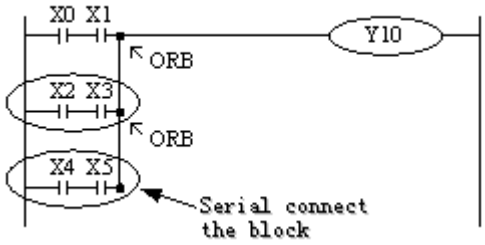
4-7.【ORB】

Mnemonic	Function	Format and device
ORB (OR Block)	Parallel connection of multiply parallel circuits	

Description

- To declare the starting point of the circuit (usually serial circuit blocks) to the preceding circuit in parallel. Serial circuit blocks are those in which more than one contacts in series or the ANB instruction is used.
- An ORB instruction is an independent instruction and is not associated with any device number.
- There are no limitations to the number of parallel circuits when using an ORB instruction in the sequential processing configuration.
- When using ORB instructions in a batch, use no more than 8 LD and LDI instructions in the definition of the program blocks (to be connected parallel).

Program



Recommended sequencial programming method：

```
LD    X0
AND   X1
LD    X2
AND   X3
ORB
LDI   X4
AND   X5
ORB
OUT   Y10
```

Non-preferred batch programming method：

```
LD    X0
AND   X1
LD    X2
AND   X3
LDI   X4
AND   X5
ORB
ORB
OUT   Y10
```

4-8.【ANB】

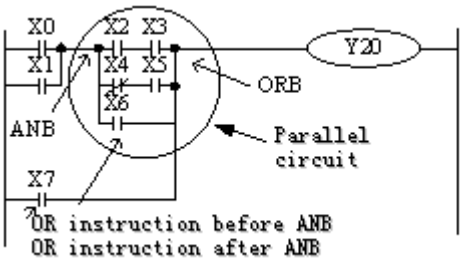
Mnemonic and Function

Mnemonic	Function	Format and device
ANB (ANd Block)	Serial connection of multiply parallel circuits	

Description

- To declare the starting point of the circuit block, use a LD or LDI instruction. After completing the parallel circuit block, connect it to the preceding block in series using the ANB instruction.
- It is possible to use as many ANB instructions as necessary to connect a number of parallel circuit blocks to the preceding block in series. When using ANB instructions in a batch, use no more than 8 LD and LDI instructions in the definition of the program blocks (to be connected in parallel)

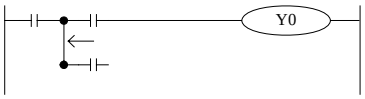
Program




```
LD      X0
OR      X1
LD      X2      Start of a branch
AND     X3
LDI     X4      Start of a branch
AND     X5
ORB             End of a parallel circuit block
OR      X6      End of a parallel circuit block
ANB             Serial connect with the preceding circuit
OR      X7
OUT     Y20
```

4-9.【MCS】,【MCR】

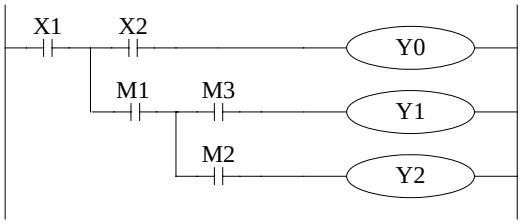
Mnemonic and Function

Mnemonic	Function	Format and device
MCS (Master control)	Denotes the start of a master control block	
MCR (Master control Reset)	Denotes the end of a master control block	

Description

- After the execution of an MCS instruction, the bus line（LD、LDI）shifts to a point after the MCS instruction. An MCR instruction returns this to the original bus line.
- MCS、MCR instructions should use in pair.
- The bus line could be used nesting. Between the matched MCS、MCR instructions use matched MCS、MCR instructions. The nest level increase with the using of MCS instruction. The max nest level is 10. When executing MCR instruction, go back to the upper bus line.
- When use flow program, bus line management could only be used in the same flow. When end some flow, it must go back to the main bus line.

Description



LD	X1	
MCS		Bus line start
OUT	Y0	
LD	M1	
MCS		Bus line nest
LD	M3	
OUT	Y1	
LD	M2	
OUT	Y2	
MCR		
MCR		Bus line back

4-10.【ALT】

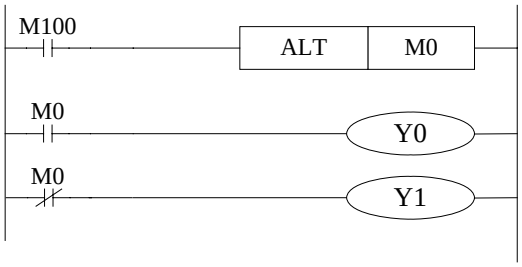
Mnemonic and Function

Mnemonic	Function	Format and device X,Y,M,S,T,C
ALT (Alternate state)	The status of the assigned devices inverted on every operation of the instruction	

Description

The status of the destination device is alternated on every operation of the ALT instruction.

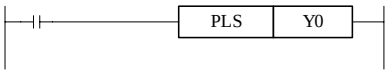
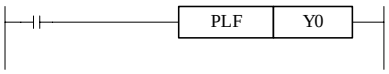
Program




```
LD      M100
ALT     M0
LD      M0
OUT     Y0
LDI     M0
OUT     Y1
```

4-11.【PLS】,【PLF】

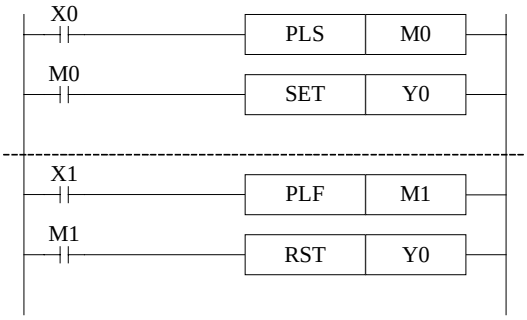
Mnemonic and Function

Mnemonic	Function	Format and device (all but special M)
PLS (PuLSe)	Rising edge pulse	
PLF (PuLse Falling)	Falling/trailing edge pulse	

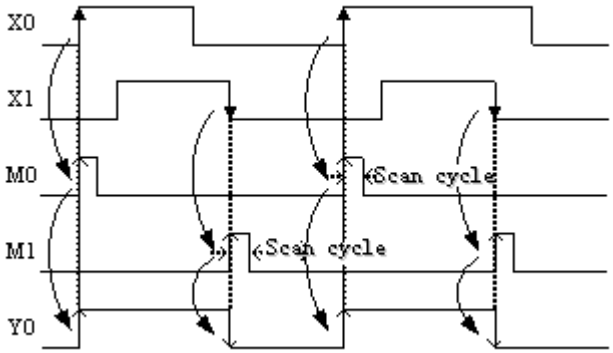
Description

- When a PLS instruction is executed, object devices Y and M operate for one operation cycle after the drive input signal has turned ON.
- When a PLF instruction is executed, object devices Y and M operate for one operation cycle after the drive input signal has turned OFF.

Program

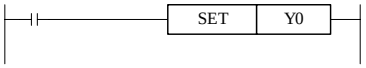
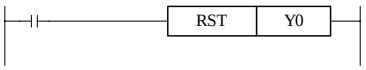



```
LD      X0
PLS     M0
LD      M0
SET     Y0
----------------------
LD      X1
PLF     M1
LD      M1
RST     Y0
```



4-12.【SET】,【RST】

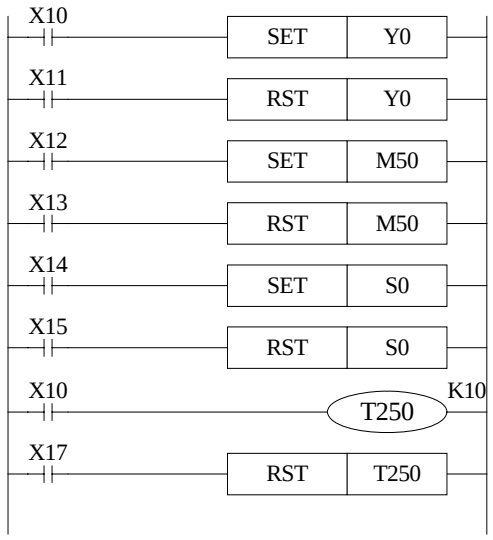
Mnemonic and Function

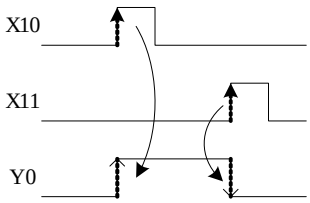
Mnemonic	Function	Format and device X,Y,M,S,T,C
SET (SET)	Set a bit device permanently ON	
RST (ReSeT)	Reset a bit device permanently OFF	

Description

- Turning ON X010 causes Y000 to turn ON. Y000 remains ON even after X010 turns OFF. Turning ON X011 causes Y000 to turn OFF. Y000 remains OFF even after X011 turns OFF. It's the same with M、S.
- SET and RST instructions can be used for the same device as many times as necessary. However, the last instruction activated determines the current status.
- After assign the start definition ID and end definition ID, operate the operands in one bound at the same time is available.
- Besides, it's also possible to use RST instruction to reset the current contents of timer, counter and contacts.
- When use SET、RST instruction, please try to avoid using the same definition ID with OUT instruction.

Program





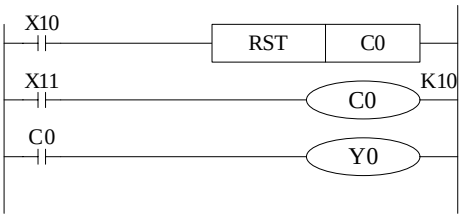

```
LD    X10
SET   Y0
LD    X11
RST   Y0
LD    X12
SET   M50
LD    X13
RST   M50
LD    X14
SET   S0
LD    X15
RST   S0
LD    X10
OUT   T250
SP    K10
LD    X17
RST   T250
```

4-13.【OUT】,【RST】for the counters

Mnemonic and Function

Mnemonic	Function	Format and device X,Y,M,S,T,C
OUT (OUT)	Final logic operation type coil drive	
RST (ReSeT)	Reset a bit device permanently OFF	

Program of interior counter

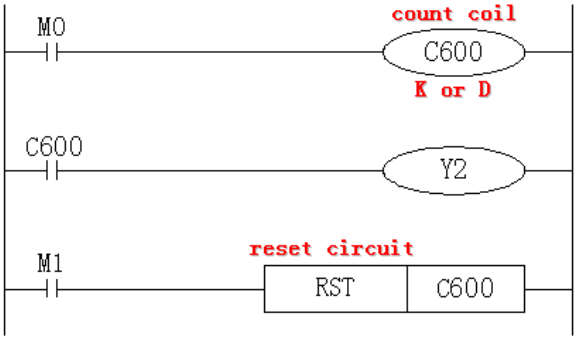


Counter used for power cut retentive. Even when power is cut, hold the current value and output contact's action status and reset status.

C0 carries on increase count for the OFF→ON of X011. When reach the set value K10, output contact C0 activates. Afterwards, even X011 turns from OFF to ON, counter's current value will not change, output contact keep on activating.

To clear this, let X010 be the activate status and reset the output contact. It's necessary to assign constant K or indirect data register's ID behind OUT instruction.

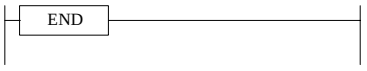
Program of high speed counter



- In the preceding example, when M0 is ON, carry on positive count with OFF→ON of X0.
- Counter's current value increase, when reach the set value (K or D), the output contact is reset.
- When M1 is ON, counter's C600 output contact is reset, counter's current value turns to be 0.

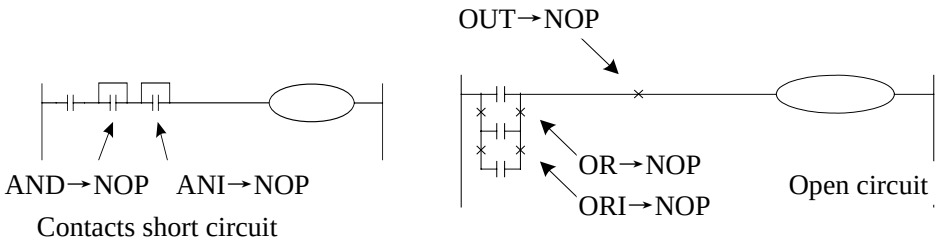
4-14.【NOP】,【END】

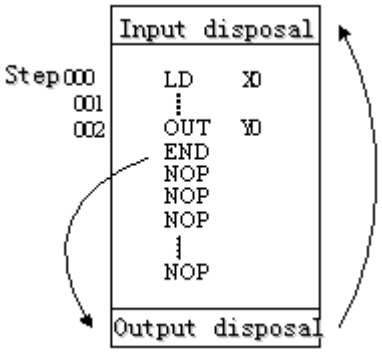
Mnemonic and Function

Mnemonic	Function	Format and device：None
NOP (No Operation)	No operation or null step	
END (END)	Force the current program scan to end	

Description

- When clear the whole program, all the instructions become NOP. If add NOP instructions between the common instructions, they have no effect and PLC will keep on working. If add NOP instructions in the program, then when modify or add programs, the step vary will be decreased. But the program should have rest quantity.
- If replace the program's instructions with NOP instructions, then the circuit will be changed, please note this.





PLC repeatedly carry on input disposal, program executing and output disposal. If write END instruction at the end of the program, then the instructions behind END instruction won't be executed. If there's no END instruction in the program, the PLC executes the end step and then repeat executing the program from step 0.

When debug, insert END in each program segment to check out each program's action.

Then, after confirm the correction of preceding block's action, delete END instruction.

Besides, the first execution of RUN begins with END instruction.

When executing END instruction, refresh monitor timer. (Check if scan cycle is a long timer.)

4-15. Items to note when programming

1、Contacts' structure and step number

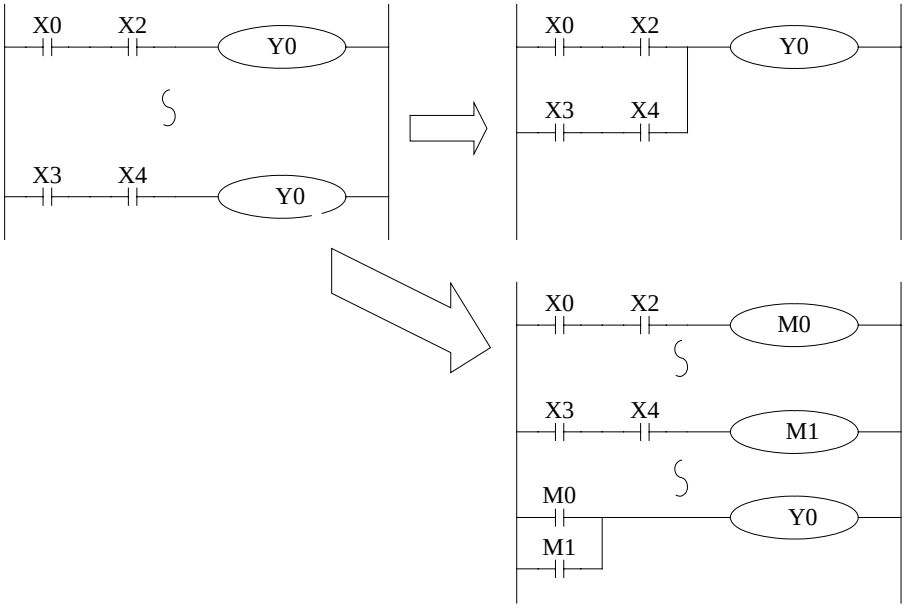
Even in the sequencial control circuit with the same action, it's also available to simple the program and save program's steps according to the contacts' structure. General program principle is：a）write the circuit with many serial contacts on the top；b）write the circuit with many parallel contacts in the left.

2、Program's executing sequence

Handle the sequencial control program by【From top to bottom】and【From left to right】Sequencial control instructions also encode following this flow.

3、Dual output dual coil's activation and the solution

- If carry on coil's dual output (dual coil) in the sequencial control program, then the backward action is prior.
- Dual output (dual coil) doesn't go against the input rule at the program side. But as the preceding action is very complicate, please modify the program as in the following example.



There are other methods. E.g. jump instructions or step ladder. However, when use step ladder, if the main program's output coil is programmed, then the disposal method is the same with dual coil, please note this.

Memo

5. Applied instruction

In this chapter, we describe applied instruction's function of XC series PLC.

5-1. Applied Instruction List

The applied instructions' sort and their correspond instructions are listed in the following table:

Common statements of XC1/XC3/XC5:

Sort	Mnemonic	Function
Program Flow	CJ	Condition jump
	CALL	Call subroutine
	SRET	Subroutine return
	STL	Flow start
	STLE	Flow end
	SET	Open the assigned flow, close the current flow
	ST	Open the assigned flow, not close the current flow
	FOR	Start of a FOR-NEXT loop
	NEXT	End of a FOR-NEXT loop
	FEND	First end
Data Compare	LD＝	LD activates if (S1) = (S2)
	LD＞	LD activates if (S1) > (S2)
	LD＜	LD activates if (S1) =< (S2)
	LD＜＞	LD activates if（S1）≠（S2）
	LD＜＝	LD activates if（S1）≤（S2）
	LD＞＝	LD activates if（S1）≥（S2）
	AND＝	AND activates if（S1）＝（S2）
	AND＞	AND activates if（S1）＞（S2）
	AND＜	AND activates if（S1）＜（S2）
	AND＜＞	AND activates if（S1）≠（S2）
	AND＜＝	AND activates if（S1）≤（S2）
	AND＞＝	AND activates if（S1）≥（S2）
	OR＝	OR activates if（S1）＝（S2）
	OR＞	OR activates if（S1）＞（S2）
	OR＜	OR activates if（S1）＜（S2）
	OR＜＞	OR activates if（S1）≠（S2）
	OR＜＝	OR activates if（S1）≤（S2）
	OR＞＝	OR activates if（S1）≥（S2）
Data Move	MOV	Move
	BMOV	Block move
	FMOV	Fill move
	FWRT	FlashROM written
	MSET	Zone set
	ZRST	Zone reset
	SWAP	The high and low byte of the destinated devices are exchanged

	XCH	Exchange
Data Operation	ADD	Addition
	SUB	Subtraction
	MUL	Multiplication
	DIV	Division
	INC	Increment
	DEC	Decrement
	MEAN	Mean
	WAND	Word And
	WOR	Word OR
	WXOR	Word exclusive OR
	CML	Compliment
	NEG	Negative

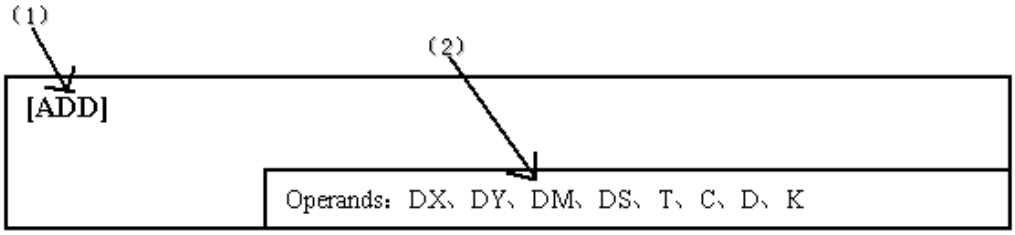
Common statements of XC3/XC5

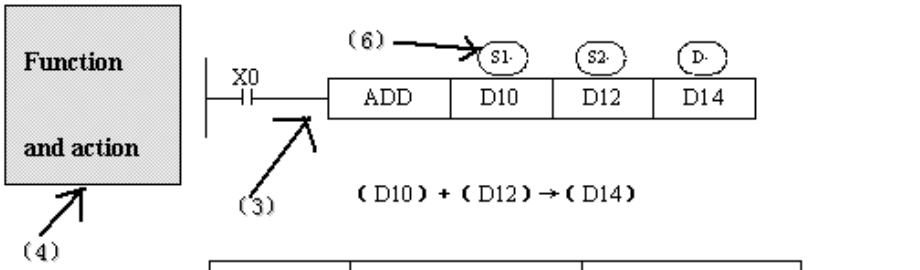
Data Shift	SHL	Arithmetic Shift Left
	SHR	Arithmetic Shift Right
	LSL	Logic shift left
	LSR	Logic shift right
	ROL	Rotation shift left
	ROR	Ritation shift right
	SFTL	Bit shift left
	SFTR	Bit shift right
	WSFL	Word shift left
	WSFR	Word shift right
Data Convert	WTD	Single word integer converts to double word integer
	FLT	32 bits integer converts to float point
	FLTD	64 bits integer converts to float point
	INT	Float point converts to binary
	BIN	BCD converts to binary
	BCD	Binary converts to BCD
	ASC	Hex. converts to ASCII
	HEX	ASCII converts to Hex.
	DECO	Coding
	ENCO	High bit coding
	ENCOL	Low bit coding
Float Point Operation	ECMP	Float compare
	EZCP	Float Zone compare
	EADD	Float Add
	ESUB	Float Subtract
	EMUL	Float Multiplication
	EDIV	Float division
	ESQR	Float Square Root
	SIN	Sine
	COS	Cosine
	TAN	Tangent
Clock Operation	TCMP	Time Compare
	TZCP	Time Zone Compare
	TADD	Time Add
	TSUB	Time Subtract
	TRD	Read RTC data
	TWR	Set RTC data

5-2. Reading method of the applied instruction's description

The understanding method of instruction's description

In this manual, instructions are described with the following format.





Flag	Zero	M8020
	Borrow bit	M8021
	Carrier	M8022

(5)

- The data contained within the two source devices are combined and the total is stored in the specified destination device. Each data's highest bit is the sign bit, 0 stands for positive,、 1 stands for negative. All calculations are algebraic processed. (5+ (-8) =-3)
- If the result of a calculation is "0", the "0" flag acts. If the result exceeds 323, 767 (16 bits limit) or 2, 147, 483, 647 (32 bits limit), the carry flag acts. (refer to the next page). If the result exceeds -323, 768 (16 bits limit) or -2, 147, 483, 648 (32 bits limit), the borrow flag acts (Refer to the next page)
- When carry on 32 bits operation, word device's low 16 bits are assigned, the device following closely the preceding device's ID will be the high bits. To avoid ID repetition, we recommend you assign device's ID to be even ID.
- The same device may be used as a source and a destination. If this is the case then the result changes after every scan cycle. Please note this point.

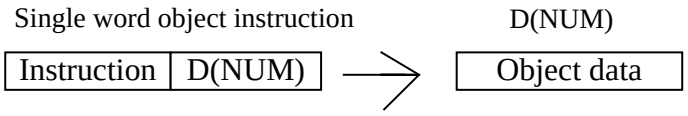
Note：

① Instruction’s name
② Device which can be used
③ Ladder example
④ Tell the instruction’s basic action, using way, applied example, extend function, note items etc.
⑤ Flag after executing the instruction. Instructions without the direct flag will not display.
⑥ (S·) : Source operand, its content won’t change after executing the instruction
(D·) : Destinate operand, its content changes with the execution of the instruction

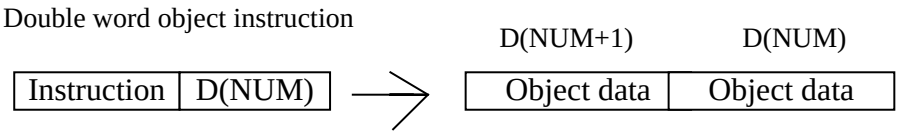
The related description

- The assignment of the data

The data register of XC series PLC is a single word (16 bit) data register, single word data only engross one data register which is assigned by single word object instruction. The disposal bound is: Dec. –327,68~327,67, Hex. 0000~FFFF.



Double word（32 bit）engrosses two data register, it's composed by two consecutive data registers, the first one is assigned by double word object instruction. The dispose bound is: Dec. -214,748,364,8~214,748,364,7, Hex. 00000000~FFFFFFFF.



- The denote way of 32 bits instruction

  If an instruction can not only be 16 bits but also be 32 bits, then the denote method for 32 bits instruction is to add a "D" before 16 bits instruction.

  E.g：ADD D0 D2 D4 denotes two 16 bits data adds；

  DADD D10 D12 D14 denotes two 32 bits data adds

Instructions contract list of **16 bits and 32 bits：**

	16 bits	32 bits
Program Flow	CJ	-
	CALL	-
	SRET	-
	STL	-
	STLE	
	SET	
	ST	
	FOR	-
	NEXT	-
	FEND	-
Data Move	MOV	DMOV
	BMOV	
	FMOV	-
	FWRT	DFWRT
	ZRST	-
	SWAP	-
	XCH	DXCH
Data operation	ADD	DADD
	SUB	DSUB
	MUL	DMUL
	DIV	DDIV
	INC	DINC
	DEC	DDEC
	MEAN	DMEAN
	WAND	DWAND
	WOR	DWOR
	WXOR	DWXOR
	CML	DCML
	NEG	DNEG
Data Shift	SHL	DSHL
	SHR	DSHR
	LSL	DLSL
	LSR	DLSR
	ROL	DROL
	ROR	DROR
	SFTL	DSFTL
	SFTR	DSFTR
	WSFL	DWSFL
	WSFR	DWSFR

	16 bits	32 bits
Data convert	WTD	-
	FLT	DFLT
	INT	DINT
	BIN	DBIN
	BCD	DBCD
	ASC	-
	HEX	-
	DECO	-
	ENCO	-
	ENCOL	-
Float operation	-	ECMP
	-	EZCP
	-	EADD
	-	ESUB
	-	EMUL
	-	EDIV
	-	ESQR
	-	SIN
	-	COS
		TAN
Clock operation	TCMP	-
	TZCP	-
	TADD	-
	TSUB	-
	TRD	-
	TWR	-

5-3. Program flow instructions

Mnemonic	Instruction's name
CJ	Condition Jump
CALL	Call subroutine
SRET	Subroutine return
STL	Flow start
STLE	Flow end
SET	Open the assigned flow, close the current flow (flow jump)
ST	Open the assigned flow, not close the current flow (Open the new flow)
FOR	Start of a FOR-NEXT loop
NEXT	End of a FOR-NEXT loop
FEND	First End

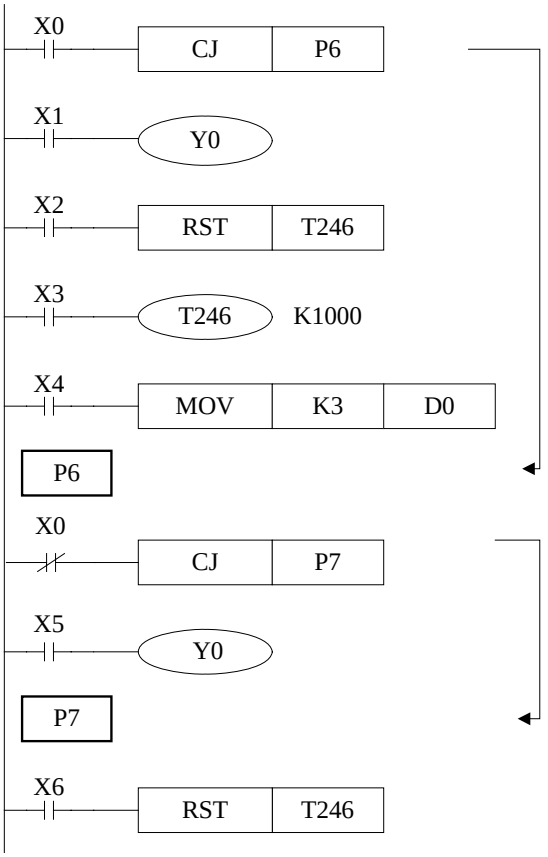
[CJ]

Operand：P

Function and Action

As the instructions of executing list, with CJ instructions, the operate cycle and dual coil can be greatly shorten.

In the following chart, if X000“ON”, then jump from step 1 to the end step of flag P6. When X000“OFF”, do not execute jump instructions.

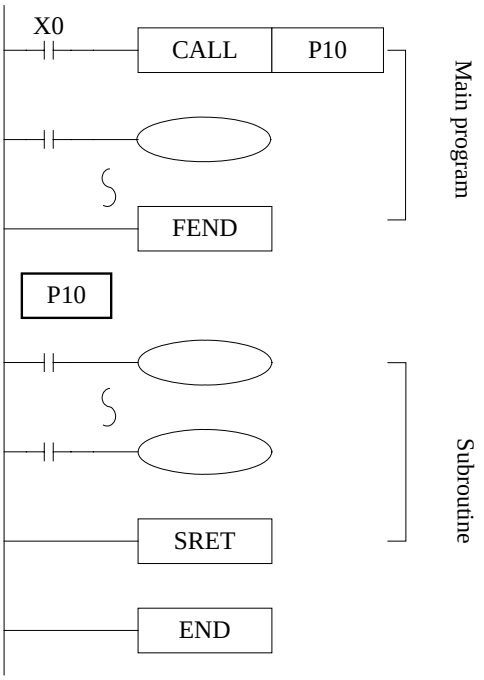


- See the upward graph, Y000 turns to be dual coil and output. But when X000=OFF, X001 activates. When X000=ON, X005 activates.
- CJ can not jump from one STL to another STL.
- If program timer T0~T640 and high speed counter C600~C640 jump after driving, go on working, output point also activate.

[CALL] and [SRET]

Operand：P

Function

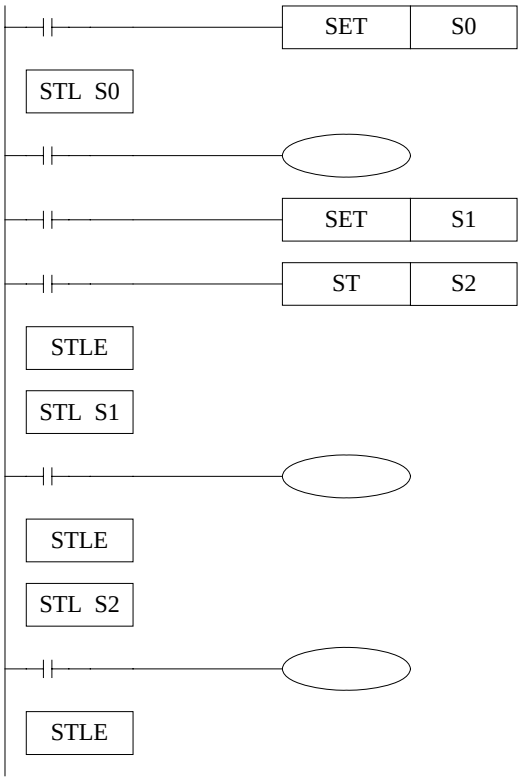


- If X000 “ON”, carry on Jump instruction and jump to step of flag P10. Here, after executing the subroutine, return to the original step via executing SRET instruction. After the following FEND instruction, program with the flag.
- In the subroutine, 9 levels Call instruction is allowed, so to the all, 10 levels nesting is available.

[SET]、[ST] and [STL]、[STLE]

Operand：S

Function



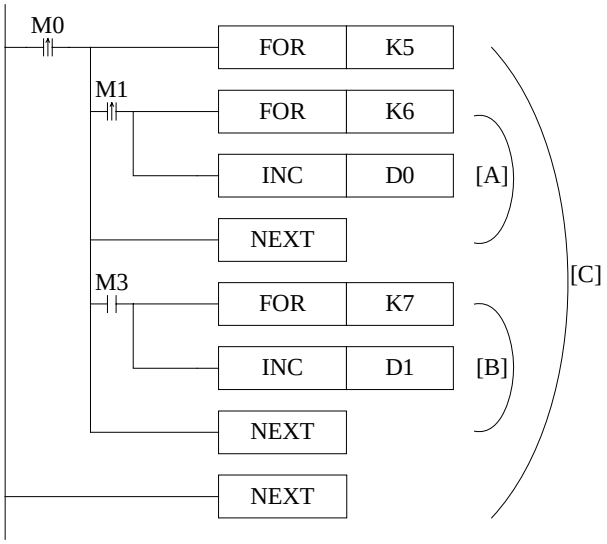
- STL and STLE should be used in pairs. STL means start of a flow, STLE means end of a flow.
- After executing of SET Sxxx instruction, the flow assigned by these instructions is ON.
- After executing RST Sxxx instruction, the assigned flow is OFF.
- In flow S0, SET S1 close the current flow S0, open flow S1.
- In flow S0, ST S2 open the flow S2, but don’t close flow S0.
- When flow turns from ON to be OFF, OFF or reset OUT、PLS、PLF、not accumulate timer etc. which belongs to the flow.
- ST instruction is usually used when a program needs to run more flows at the same time.

[FOR] and [NEXT]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function

First execute the instructions between FOR~NEXT instructions for several times(the loop time is assigned by the source data), then execute the steps after NEXT.

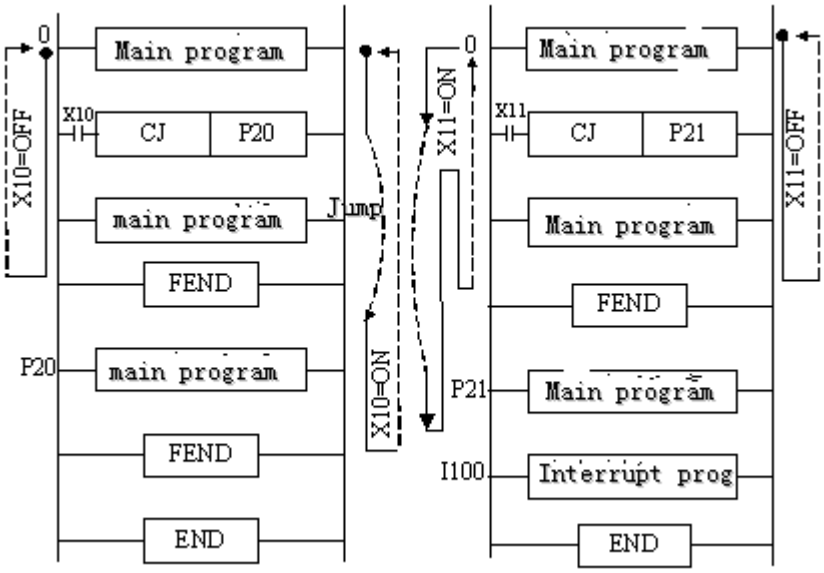


- FOR、NEXT instructions must be programmed as a pair. Nesting is allowed, and the nesting level is 8.
- Between FOR/NEXT, LDP、LDF instructions are effective for one time. Everytime when M0 turns from OFF to ON, and M1 turns from OFF to ON, [A] loop is executed 6 times.
- Everytime if M0 turns from OFF to ON and M3 is ON, [B] loop is executed 5×7=35 times.
- If there are many loop times, the scan cycle will be prolonged. Monitor timer error may occur, please note this.
- If NEXT is before FOR, or no NEXT, or NEXT is behind FENG，END, or FOR and NEXT number is not equal, an error will occur.
- Between FOR~NEXT, CJ nesting is not allowed, also in one STL, FOR~NEXT must be programmed as a pair.

[FEND] and [END]	
	Operand：None

Function

An FEND instruction indicates the first end of a main program and the start of the program area to be used for subroutines. Under normal operating circumstances the FEND instruction performs a similar action to the END instruction, i.e. output processing, input processing and watchdog timer refresh are all carried out on execution.



- If program the tag of CALL instruction behind FEND instruction, there must be SRET instruction. If the interrupt pointer program behind FEND instruction, there must be SRET instruction.
- After executing CALL instruction and before executing SRET instruction, if execute FEND instruction; or execute FEND instruction after executing FOR instruction and before executing NEXT, then an error will occur.
- In the condition of using many FEND instruction, please compile routine or subroutine between the last FEND instruction and END instruction.

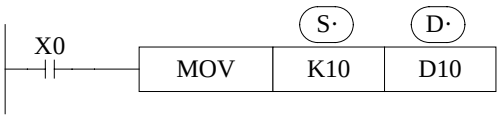
5-4. Data Move

Mnemonic	Function
MOV	Move
BMOV	Block Move
FMOV	Fill Move
FWRT	Written of FlashROM
ZRST	Zone Reset
SWAP	Float To Scientific
XCH	Exchange

[MOV]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function



Move data from one storage area to a new.

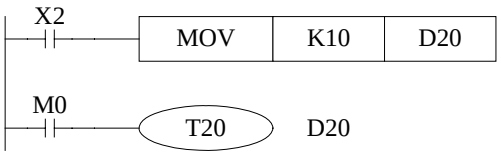
- Move contents from source to destination
- If X000 is OFF, data will not change.
- Constant K10 will automatically convert to be BIN code.

《Read out the current value of timer、counter》

（T0 current value）→（D20）

It's the same with the counter.

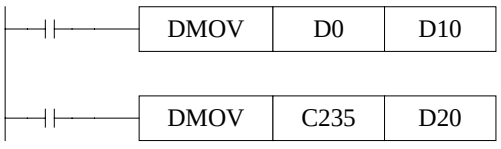
《Indirect assign the set value of timer、counter》



（K10）（D10）

D20=K10

《Move of 32 bits data》



（D1，D0）→（D11，D10）

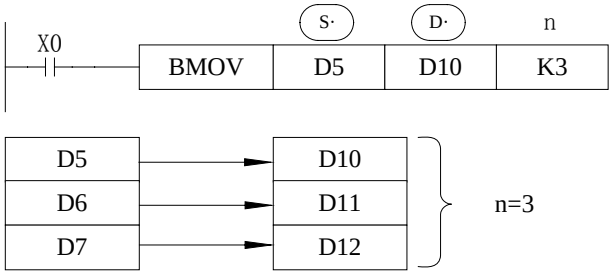
（C235 current value）→（D21，D20）

[BMOV]

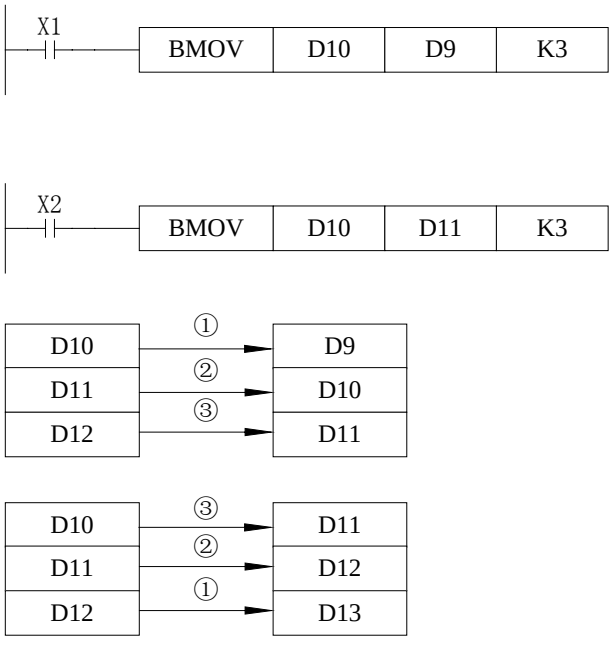
Operands：DX、DY、DM、DS、T、C、D、FD、K

Function

- A quantity of consecutively occurring data elements can be copied to a new destination. The source data is identified as a device head address(S) and a quantity of consecutive data elements (n). This is moved to the destination device (D) for the same number of elements (n). (If the quantity of source device (n) exceeds the actual number of available source devices, then only those devices which fall in the available range will be used. If the number of source devices exceeds the available space at the destination location, then only the available destination devices will be written to.)



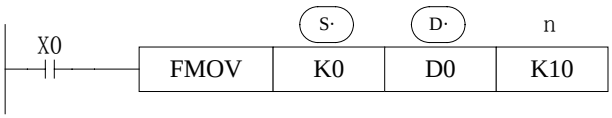
- The BMOV instruction has a built in automatic feature to prevent overwriting errors from occurring when the source (S-n) and destination (D-n) data ranges coincide. This is clearly identified in the following diagram:
- (NOTE: The numbered arrows indicate the order in which the BMOV is processed).



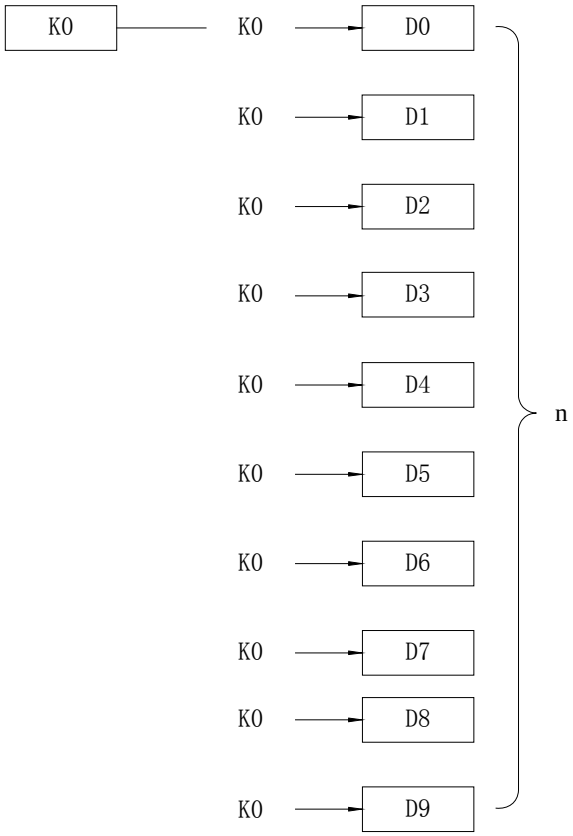
[FMOV]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function



- Move K0 to D0~D9. Copy a single data device to a range of destination devices.
- The data stored in the source device (S) is copied to every device within the destination range, The range is specified by a device head address (D) and a quantity of consecutive elements (n).
- If the specified number of destination devices (n) exceeds the available space at the destination location, then only the available destination devices will be written to.

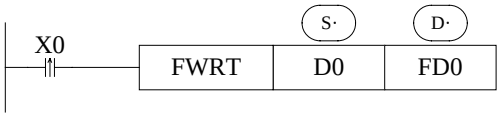


[FWRT]

Operands：DX、DY、DM、DS、T、C、D、FD、K

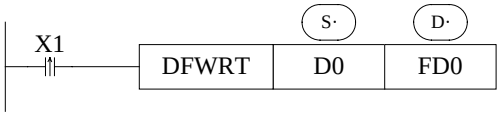
Function

1、Written of a word



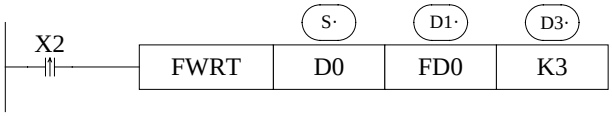
Function：write value in D0 into FD0

2、Written of double word



Function：write value in D0、D1 into FD0、FD1

3、Written of multi-word



Function：write value in D0、D2、D3 into FD0、FD1、FD2.

Note：1、FWRT instruction only allow to write data into FlashROM register. In this storage area, even battery drop, data could be stored. So it could be used to store important technical parameters.

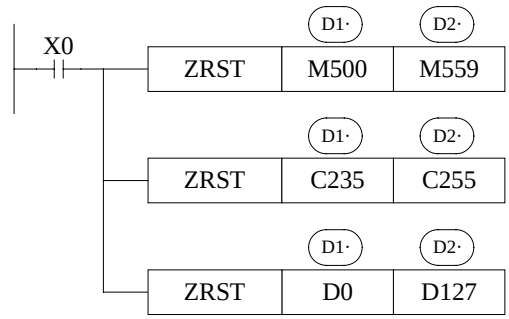
2、Written of FWRT needs a long time, about 150ms, so, frequently operate this operation is not recommended.

3. The written time of FlashROM is about 1,000,000 times. So, we suggest using edge signals (LDP、LDF etc.) to trigger.

[ZRST]

Operands：All bit、word soft units

Function



Zone reset bit device M500~M599。

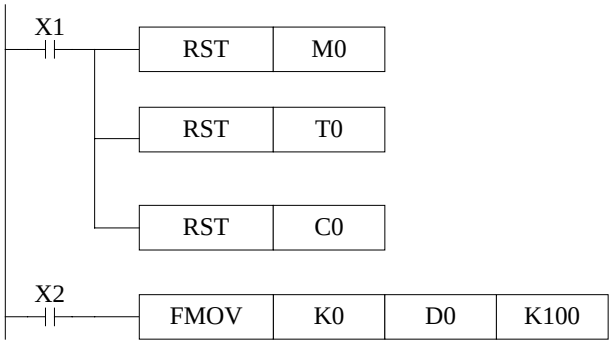
Zone reset word device C235~C255

Zone reset status S0~S127

- D1 and D2 are assigned to be the same device, and D1 < = D2. When D1>D2, only reset device in D1.
- The instruction is 16 bits, but it's available to use D1, D2 to assign 32 bits counter. But mix assignment is not allowed. I.e. D1 is a 16 bits counter, D2 is a 32 bits counter, this condition is not allowed.

Other reset instructions

- As single reset instruction of device, RST instruction is available of bit device Y，M，S and word device T，C，D.
- As Fill Move instruction of K0, you could write 0 into device DX，DY，DM，DS，T，C，D.



Reset M0

Reset the current value of T0

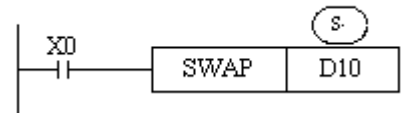
Reset D0

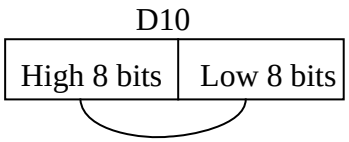
Write K0 into D0~D9

[SWAP]

Operands：DX、DY、DM、DS、T、C、D 、FD

Function





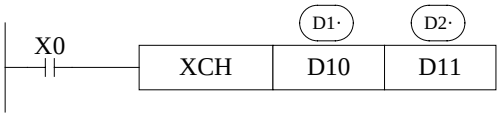
- Low 8 bits and high 8 bits change when it is 16 bits instruction.
- If the instruction is a consecutive executing instruction, each operation cycle should change.

[XCH]

Operands：DX、DY、DM、DS、T、C、D、FD

Function

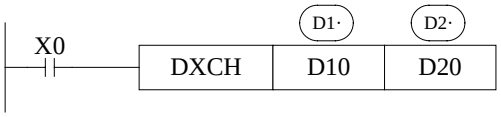
《16 bits instruction》



Before（D10）=100 →After （D10）=101
（D11）=101 （D11）=100

- The contents of the two destination devices D1 and D2 are swapped,
- When drive input X0 is ON, each scan cycle should carry on data exchange, please note.

《32 bits instruction》



- 32 bits instruction [DXCH] swaps value composed by D10、D11 and the value composed by D20、D21.

5-5. Data operation instructions

Mnemonic	Function
ADD	Addition
SUB	Subtraction
MUL	Multiplication
DIV	Division
INC	Increment
DEC	Decrement
MEAN	Mean
WAND	Logic Word And
WOR	Logic Word Or
WXOR	Logic Exclusive Or
CML	Compliment
NEG	Negation

[ADD]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function

（D10）＋（D12）→（D14）

Flag	Zero	M8020
	Borrow	M8021
	Carry	M8022

- The data contained within the two source devices are combined and the total is stored in the specified destination device. Each data's highest bit is the sign bit, 0 stands for positive,、1 stands for negative. All calculations are algebraic processed.（5+（-8）=-3）
- If the result of a calculation is "0", the "0" flag acts. If the result exceeds 323，767（16 bits limit）or 2，147，483，647（32 bits limit）, the carry flag acts.（refer to the next page）. If the result exceeds -323，768（16 bits limit）or -2，147，483，648（32 bits limit）, the borrow flag acts（Refer to the next page）
- When carry on 32 bits operation, word device's low 16 bits are assigned, the device following closely the preceding device's ID will be the high bits. To avoid ID repetition, we recommend you assign device's ID to be even ID.
- The same device may be used as a source and a destination. If this is the case then the result changes after every scan cycle. Please note this point.

[SUB]

Operands：DX、DY、DM、DS、T、C、D、FD、K

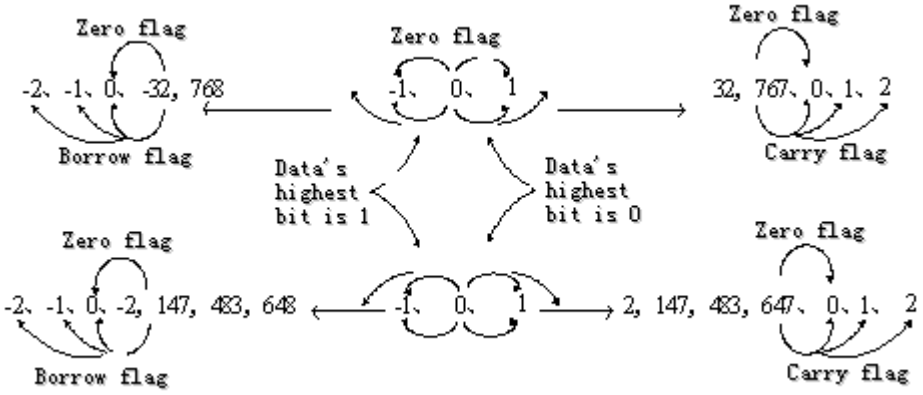
Function

（D10）—（D12）→（D14）

Flag	Zero	M8020
	Borrow bit	M8021
	Carrier	M8022

- (S1·) appoint the soft unit's content, subtract the soft unit's content appointed by (S2·) in the format of algebra. The result will be stored in the soft unit appointed by (D·). (5-(-8)=13)
- The action of each flag, the appointment method of 32 bits operation's soft units are both the same with the preceding ADD instruction.

The relationship of the flag's action and positive/negative data is the following chart.

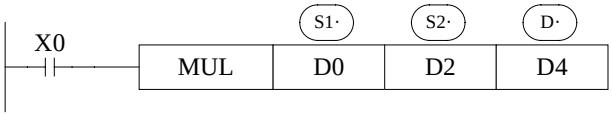


[MUL]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function and action

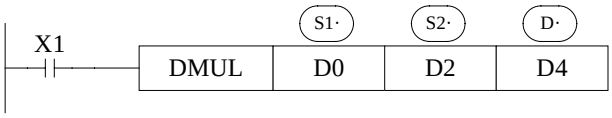
《16 bits operation》



BIN		BIN		BIN
（D0）	×	(D2)	→	（D5，D4）
16 bits		16 bits	→	32 bits

- The contents of the two source devices are multiplied together and the result is stored at the destination device in the format of 32 bits. As in the upward chart： when （D0）=8、（D2）=9，（D5，D4）=72。
- The result's highest bit is the symbol bit: positive（0）、negative（1）.
- When be bit unit, it can carry on the bit appointment of K1~K8. When appoint K4, only the result's low 16 bits can be obtained.

《32 bits operation》



BIN		BIN		BIN
D1，D0）	×	(D3，D2)	→	（D7，D6，D5，D4）
32 bits		32 bits	→	64 bits

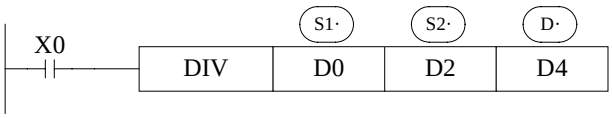
- In 32 bits operation, when use bit device as the destination address, only low 32 bits result can be obtained. The high 32 bits result can not be obtained, so please operate again after transfer one time to the word device
- Even use word device, 64 bits results can't be monitored at once.
- In this situation, float point data operation is recommended.

[DIV]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function and action

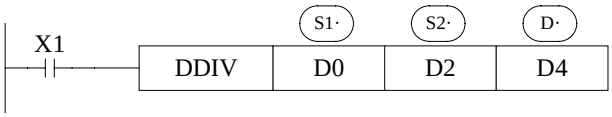
《16 bits operation》



Dividend		Divisor		Result		Remainder
BIN		BIN		BIN		BIN
(D0)	÷	(D2)	→	D4）	---	(D5)
16 bits		16 bits		16 bits		6 bits

- (S1·) appoints the device's content be the dividend, (S2·) appoints the device's content be the divisor, (D·) appoints the device and the next one to store the result and the remainder.

《32 bits operation》



Dividend		Divisor	Result		Result
BIN		BIN	BIN		BIN
(D1,D0)	÷	(D3,D2)	（D5,D4）	---	(D7,D6)
32 bits		32 bits	32 bits		32 bits

- The dividend is composed by the device appointed by (S1·) and the next one. The divisor is composed by the device appointed by (S2·) and the next one. The result and the remainder are stored in the four sequencial devices, the first one is appointed by (D·).
- If the value of the divisor is 0, then an operation error is executed and the operation of the DIV instruction is cancelled.
- When appoint the bit device as (D·), the remainder will not obtained.
- The highest bit of the result and remainder is the symbol bit (positive:0, negative: 1). When any of the dividend or the divisor is negative, then the result will be negative. When the dividend is negative, then the remainder will be negative.

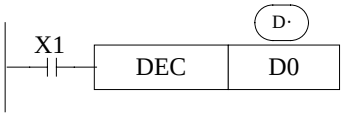
[INC] and [DEC]

Operands：DX、DY、DM、DS、T、C、D、FD

Function and action

（D0）＋1→(D0)

- On every execution of the instruction the device specified as the destination (D·) has its current value incremented (increased) by a value of 1.
- In 16 bits operation, when +32，767 is reached, the next increment will write -32，767 to the destination device. In this case, there's no additional flag to identify this change in the counted value.



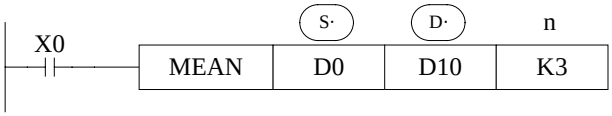
（D0）－1→(D0)

- On every execution of the instruction the device specified as the destination (D·) has its current value decremented (decreased) by a value of 1.
- When -32，768 or -2，147，483，648 is reached, the next decrement will write +32，767 or +2，147，483，647 to the destination device.

[MEAN]

Operands：DX、DY、DM、DS、T、C、D、FD

Function



$$\frac{(D0) + (D1) + (D2)}{3} \longrightarrow (D10)$$

- The value of all the devices within the source range is summed and then divided by the number of devices summed, i.e. n.. This generates an integer mean value which is stored in the destination device (D) The remainder of the calculated mean is ignored.
- If the value of n is specified outside the stated range (1 to 64) an error is generated.

[WAND]、[WOR] and [WXOR]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function

- Execute logic AND operation with each bit

0&0=0　0&1=0
1&0=0　1&1=1

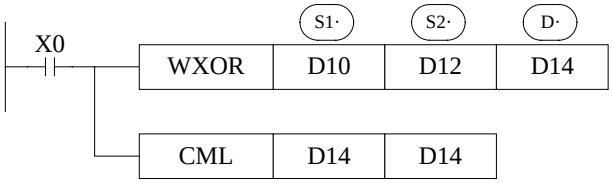
- Execute logic OR operation with each bit

0 or 0=0　0 or 1=1
1 or 0=1　1 or 1=1

- Execute logic Exclusive OR operation with each bit.

0 xor 0=0　0 xor 1=1
1 xor 0=1　1 xor 1=0

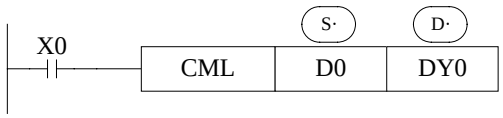
If use this instruction along with CML instruction, XOR NOT operation could also be executed .



[CML]

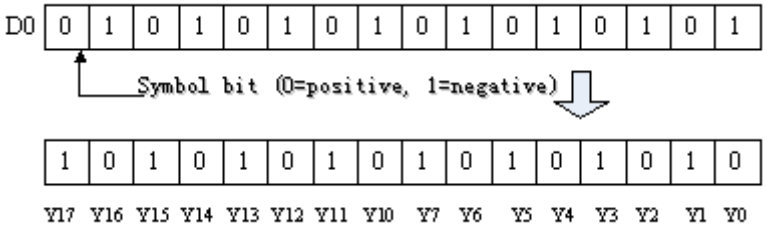
Operands：DX、DY、DM、DS、T、C、D、FD

Function

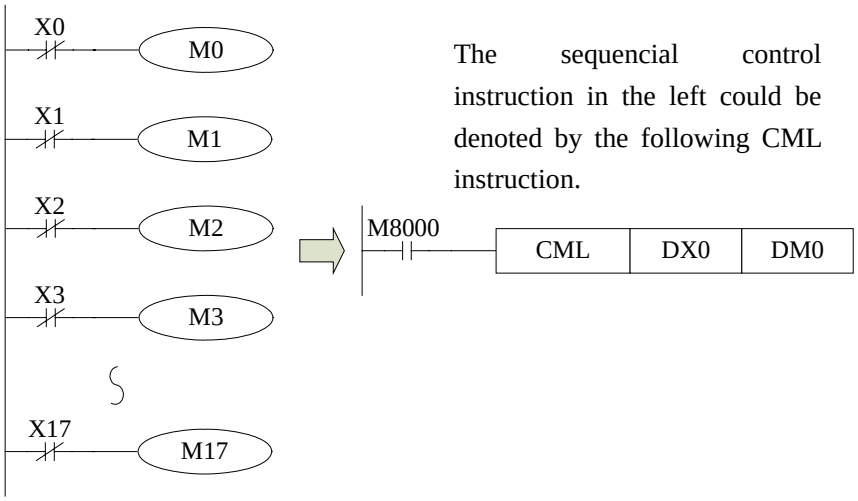


A copy of each data bit within the source device (S·) is inverted and then moved to the designated destination (D·).

- Each data bit in the source device is inverted and sent to the destination device. If use constant K in the source device, it can be auto convert to be binary.
- It's available when you want to inverted output the PLC's output



《Reading of inverted input》

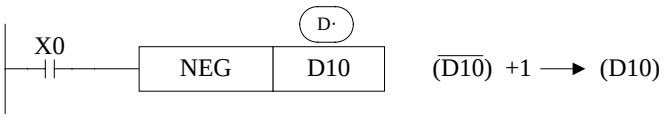


The sequencial control instruction in the left could be denoted by the following CML instruction.

NEG

Operands：DX、DY、DM、DS、T、C、D、FD

Function and action



- The bit format of the selected device is inverted, I.e. any occurrence of a “1’ becomes a “0” and any occurrence of “0” becomes “1”, when this is complete, a further binary 1 is added to the bit format. The result is the total logic sigh change of the selected devices contents.
- When using continually executing instructions, then this instruction will be executed in every scan cycle.

5-6. Shift instructions

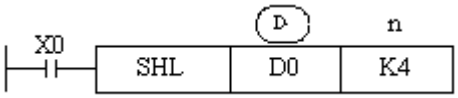
Mnemonic	Function
SHL	Arithmetic shift left
SHR	Arithmetic shift right
LSL	Logic shift left
LSR	Logic shift right
ROL	Rotation left
ROR	Rotation right
SFTL	Bit shift left
SFTR	Bit shift right
WSFL	Word shift left
WSFR	Word shift right

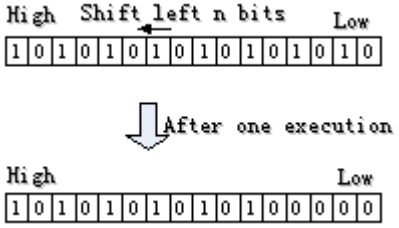
[SHL] and [SHR]

Operands：DX、DY、DM、DS、T、C、D、FD

Function and action

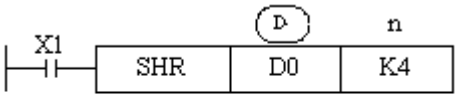
《Arithmetic shift left》

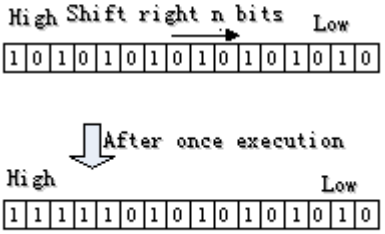




- After one execution, fill 0 in the low bit

《Arithmetic shift left》





- After once execution, the highest bit remains.

NOTE:

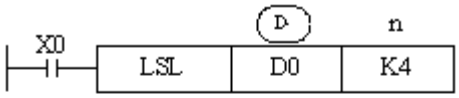
- In every scan cycle, loop shift left/right action will be executed
- The situation of 32 bits is the same.

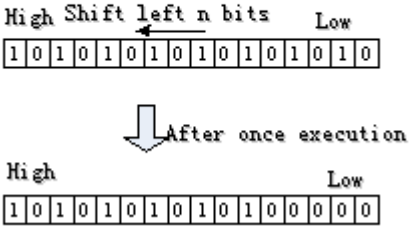
[LSL] and [LSR]

Operands：DX、DY、DM、DS、T、C、D、FD

Function and action

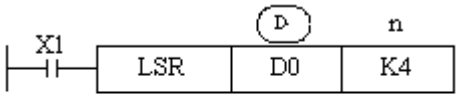
《Logic shift left》

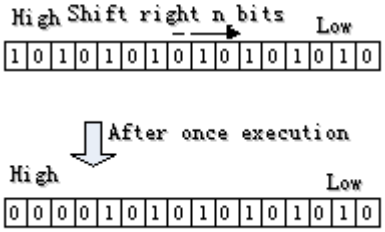




- After once execution, fill 0 in the low bit.

《Logic shift right》





- After once execution, fill 0 in the highest bit.

NOTE:

- In every scan cycle, loop shift left/right action will be executed
- The situation of 32 bits is the same.

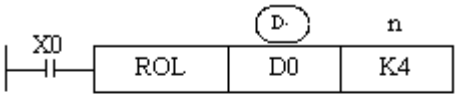
[ROL] and [ROR]

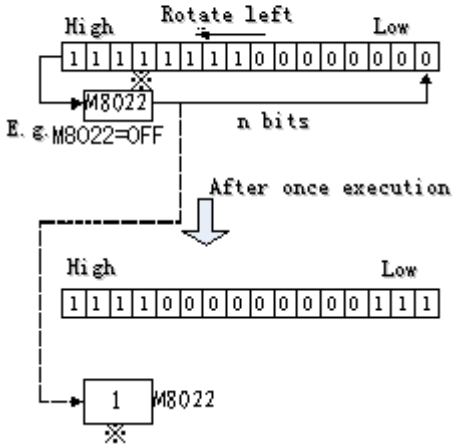
Operands：DX、DY、DM、DS、T、C、D、FD

Function and action

The bit format of the destination device is rotated n bit places to the left on every operation of the instruction

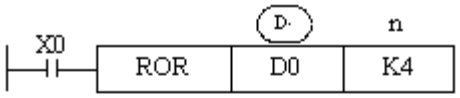
《Rotation shift left》

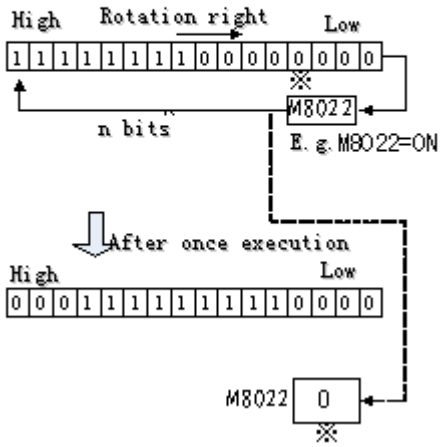




- Every time when X000 turns from OFF to ON, executes n bits left rotation.

《Rotation shift right》





- Every time when X000 turns from OFF to ON, executes n bits right rotation.

- As there is a carry flag in the rotation circuit, so if drive M8022 before executing the rotation instruction, it could be sent to the destination address.
- Please note that rotation left/right action is executed in every scan cycle.
- The situation of 32 bits is the same.

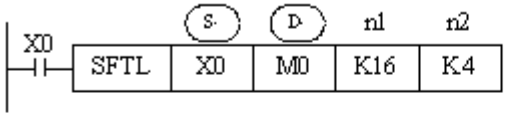
[SFTL] and [SFTR]

Operands：DX、DY、DM、DS、T、C、D、FD

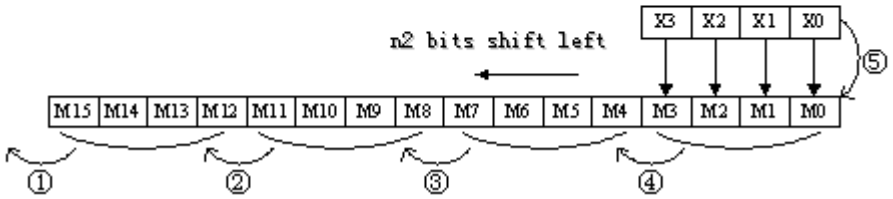
Function and action

- The instruction copies n2 source devices to a bit stack of length n1. For every new addition of n2 bits, the existing data within the bit stack is shifted n2 bits to the left/right. Any bit data moving to the position exceeding the n1 limit is diverted to an overflow area. The bit shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

《Bit shift left》

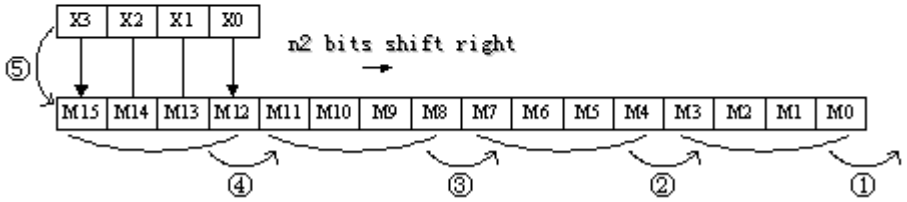


① M15~M12→overflow
② M11~M 8→M15~M 12
③ M 7~M 4→M11~M8
④ M 3~M 0→M7~M4
⑤ X 3~X 0→M3~M0



《Bit shift right》

① M 3~M 0→overflow
② M 7~M 4→M3~M0
③ M11~M 8→M7~M4
④ M15~M12→M11~M8
⑤ X 3~X 0→M15~M12



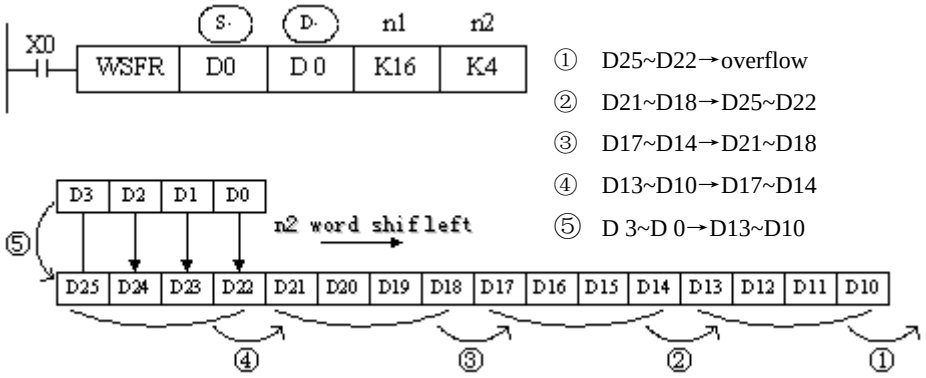
[WSFL] and [WSFR]

Operands：DX、DY、DM、DS、T、C、D、FD

Function and action

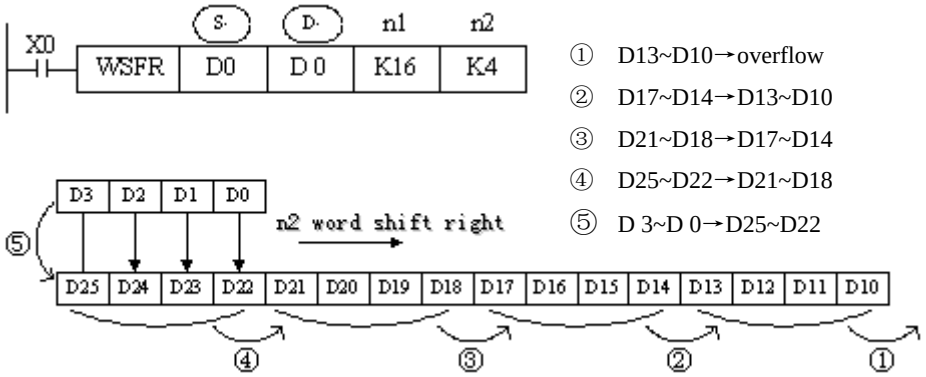
- The instruction copies n2 source devices to a word stack of length n1. For each addition of n2 words, the existing data within the word stack is shifted n2 words to the left/right. Any word data moving to a position exceeding the n1 limit is diverted to an overflow area. The word shifting operation will occur everytime the instruction is processed unless it is modified with either the pulse suffix or a controller interlock.

《Word shift left》



① D25~D22→overflow
② D21~D18→D25~D22
③ D17~D14→D21~D18
④ D13~D10→D17~D14
⑤ D 3~D 0→D13~D10

《Word shift right》



① D13~D10→overflow
② D17~D14→D13~D10
③ D21~D18→D17~D14
④ D25~D22→D21~D18
⑤ D 3~D 0→D25~D22

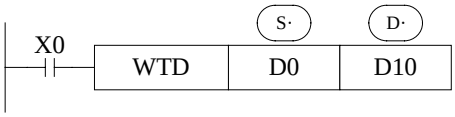
5-7. Data Convert

Mnemonic	Function
WTD	Single word integer converts to double word integer
FLT	32 bits integer converts to float point
FLTD	64 bits integer converts to float point
INT	Float point converts to integer
BIN	BCD convert to binary
BCD	Binary converts to BCD
ASC	Hex. converts to ASCII
HEX	ASCII converts to Hex.
DECO	Coding
ENCO	High bit coding
ENCOL	Low bit coding

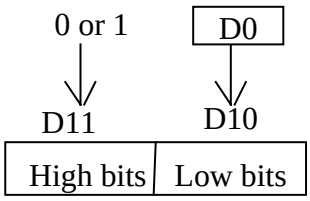
[WTD]

Operands：DX、DY、DM、DS、T、C、D、FD

Function



（D0） → （D11，D10）
Single word Double word



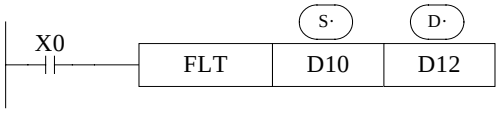
- When single word D0 is positive integer, after executing this instruction, the high bit of double word D10 is 0.
- When single word D0 is negative integer, after executing this instruction, the high bit of double word D10 is 1.

[FLT] and [FLTD]

Operands：DX、DY、DM、DS、T、C、D、FD

Function

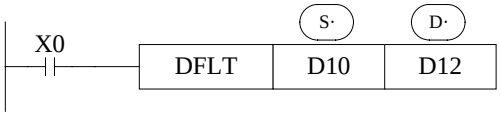
《16 bits》



(D10) → (D13,D12)

BIN integer Binary float point

《32 bits》



（D11,D10）→ （D13,D12）

BIN integer Binary float point

《64 bits》

（D13,D12,D11,D10）→ （D17,D16,D15,D14）

BIN integer Binary float point

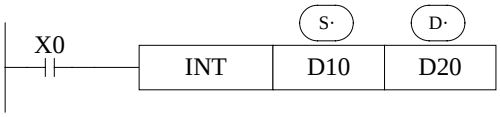
- Convert BIN integer to binary float point. As the constant K、H will auto convert by the float operation instruction, so this FLT instruction can’t be used.
- The instruction is contrary to INT instruction.

[INT]

Operands：DX、DY、DM、DS、T、C、D、FD

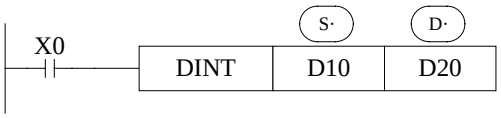
Function

《16 bits》



（D11,D10） → (D20)

《32 bits》



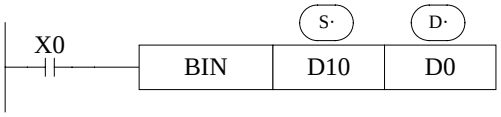
（D11,D10） → (D20)

- The binary source number is converted into an BIN integer and stored at the destination device. Abandon the value behind the decimal point.
- This instruction is contrary to FLT instruction.
- When the result is 0, the flag bit is ON。
  When converting, less than 1 and abandon it, zero flag is ON.
  16 bits operation：-32,768~32,767
  32 bits operation：-2,147,483,648~2,147,483,647

[BIN]

Operands：DX、DY、DM、DS、T、C、D、FD

Function



Data's bound：0~9,999 or 0~99，999，999 is valid.

Convert and move instruction of Source（BCD）→ destination（BIN）

- When source data is not BCD code, M8067（Operation error）， M8068（Operation error lock）will not work.
- As constant K automatically converts to binary, so it's not suitable for this instruction.

[BCD]

Operands：DX、DY、DM、DS、T、C、D、FD

Function

Convert and move instruction of source (BIN)→destination (BCD).

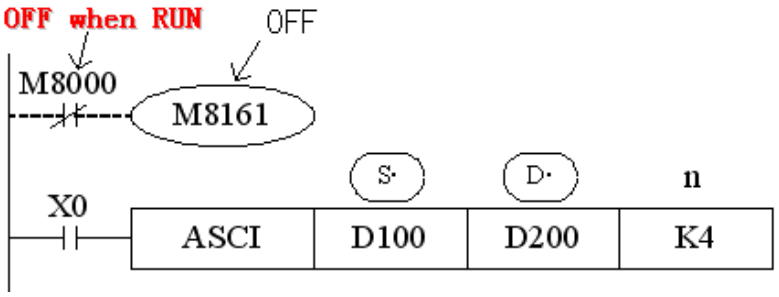
- When use BCD instruction, if the converted BCD number exceeds the operational ranges of 0 to 9999(16 bits operation) and 0 to 99999999 (32 bit operation) an error will occur.
- This instruction can be used to output data directly to a seven segment display.

[ASCI]

Operands: DX、DY、DM、DS、T、C、D、FD

Function and action

《16 bits convert mode》When M8161=OFF



Convert each bit of source's (S) Hex. format data to be ASCII code, move separately to the high 8 bits and low 8 bits of destination (D). The convert alphanumeric number is assigned with n.
(D) is low 8 bits, high 8 bits, store ASCII data.

The convert result is the following：

Assign start device：	[0]=30H	[1]=31H	[5]=35H
(D100)=0ABCH	[A]=41H	[2]=32H	[6]=36H
(D101)=1234H	[B]=42H	[3]=33H	[7]=37H
(D102)=5678H	[C]=43H	[4]=34H	[8]=38H

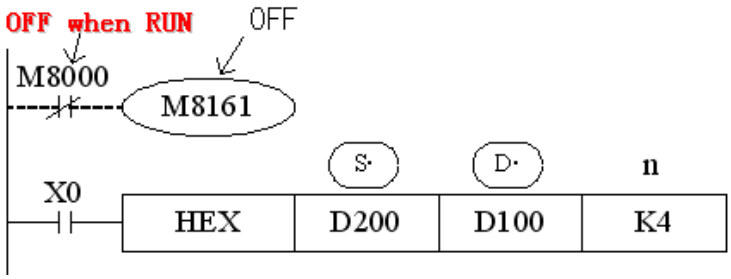
D \ n	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200 down	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D200 down		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D201 down			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D201 up				[C]	[B]	[A]	[0]	[4]	[3]
D202 down					[C]	[B]	[A]	[0]	[4]
D202 up						[C]	[B]	[A]	[0]
D203 down							[C]	[B]	[A]
D203 up								[C]	[B]
D204 down									[C]

[HEX]

Operands：DX、DY、DM、DS、T、C、D、FD

Function and action

《16 bits switch mode》 When M8161=OFF

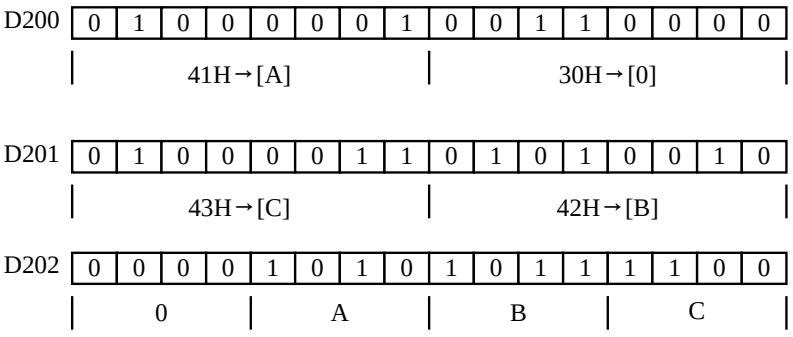


Convert the high and low 8 bits in source to HEX data. Move 4 bits every time to destination. The convert alphanumeric number is assigned by n.

The convert of the upward program is the following：

(S・)	ASCII	HEX Conversion
D200 up	30H	0
D200 up	41H	A
D201 down	42H	B
D201 up	43H	C
D202 down	31H	1
D202 up	32H	2
D203 down	33H	3
D203 up	34H	4
D204 down	35H	5

n \ (D・)	D102	D101	D100
1	Not change to be 0		••• 0H
2			•• 0AH
3			• 0ABH
4			0ABCH
5		••• 0H	ABC1H
6		•• 0AH	BC12H
7		• 0ABH	C123H
8		0ABCH	1234H
9	••• 0H	ABC1H	2345H

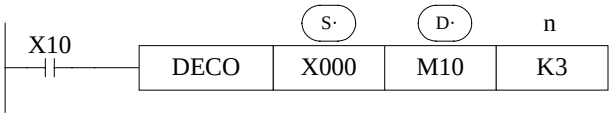


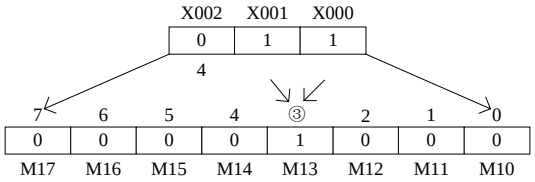
[DECO]

Operands：DX、DY、DM、DS、T、C、D、FD

功能和动作

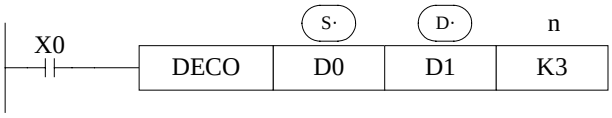
《When (D·) is software unit》 n≤16

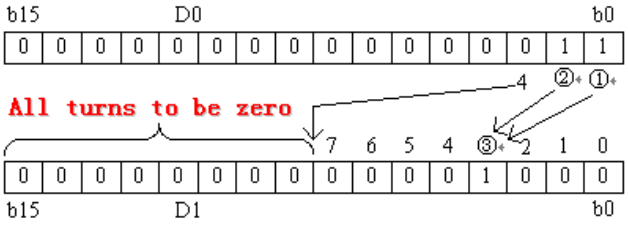




- The source address is 1+2=3，so starts from M10, the number 3 bit (M13) is 1. If the source are all 0, M10 is 1
- When n=0, no operation, beyond n=0~16, don't execute the instruction.
- When n=16, if coding command "D" is soft unit, it's point is 2^8=256。
- When drive input is OFF, instructions are not executed, the activate coding output keep on activate.

《When (D·) is word device》 n≤4





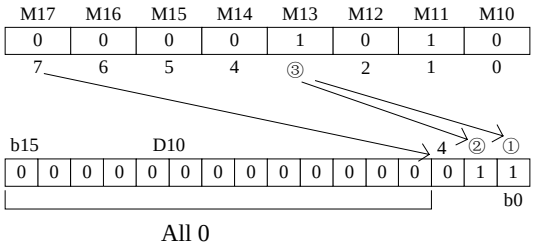
- Source ID's low n bits（n≤4） are encoded to the destination ID. When n ≤3, destination's high bits all converts to be 0.
- When n=0, no disposal, beyond n=0~4, don't execute the instruction.

[ENCO]

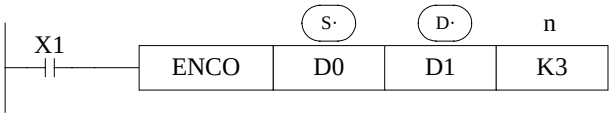
Operands：DX、DY、DM、DS、T、C、D、FD

Function

《When (S·) is bit device》 n≤16



《When (S·) is word device》 n≤16



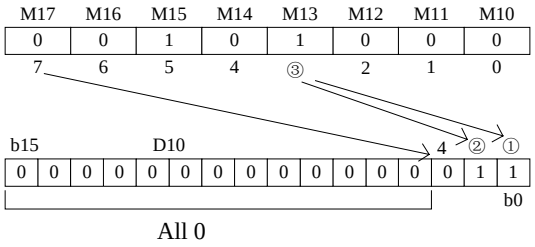
- If many bits in the source ID are 1, ignore the low bits. If source ID are all 0, don't execute the instructions.
- When drive input is OFF, the instruction is not executed, encode output don't change.
- When n=8, if encode instruction's "S" is bit unit, it's point number is 2^8=256

[ENCOL]

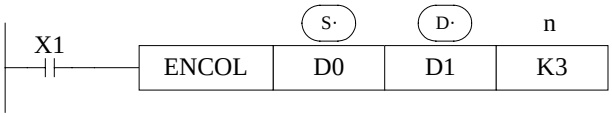
Operands：DX、DY、DM、DS、T、C、D、FD

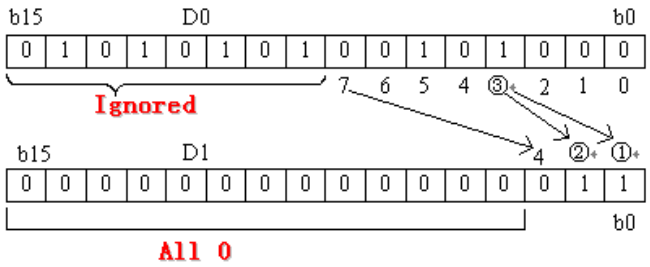
Function

《If (S·) is bit device》 n≤16



《If (S·) is word device》 n≤16





- If many bits in the source ID are 1, ignore the high bits. If source ID are all 0, don't execute the instructions.
- When drive input is OFF, the instruction is not executed, encode output don't change.
- When n=8, if encode instruction's "S" is bit unit, it's point number is 2^8=256

5-8. Floating Operation

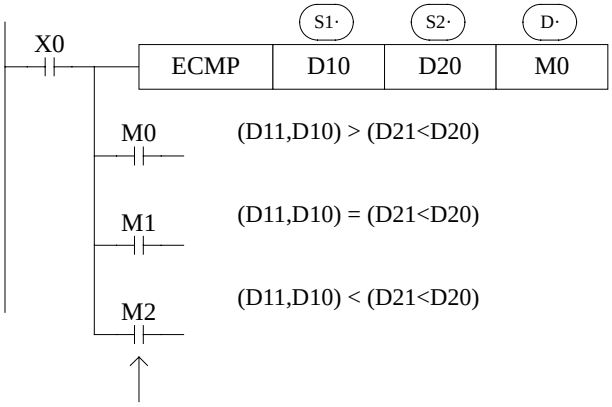
Mnemonic	Function
ECMP	Float Compare
EZCP	Float Zone Compare
EADD	Float Add
ESUB	Float Subtract
EMUL	Float Multiplication
EDIV	Float Division
ESQR	Float Square Root
SIN	Sine
COS	Cosine
TAN	Tangent

[ECMP]

Operands：DX、DY、DM、DS、T、C、D、FD、K

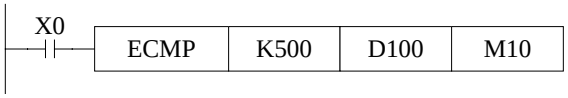
Function and action

（D11,D10） ： （D21,D20）→M0,M1,M2



The status of the destination device will be kept even if the ECMP instruction is deactivated.

- The binary float data of S1 is compared to S2. The result is indicated by 3 bit devices specified with the head address entered as D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



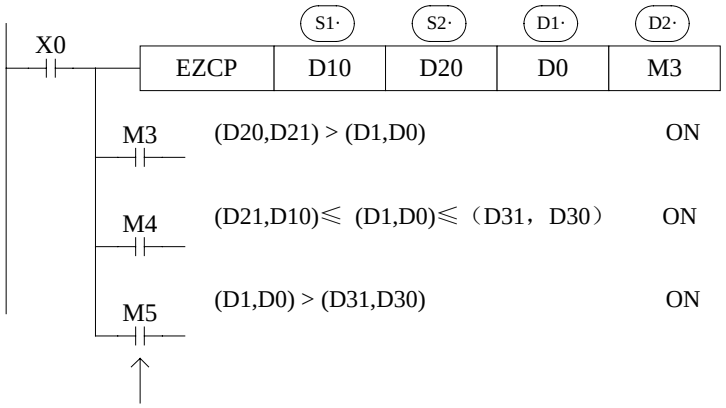
（K500）：（D101，D100）→M10,M11,M12

[EZCP]

Operands：DX、DY、DM、DS、T、C、D、FD、K

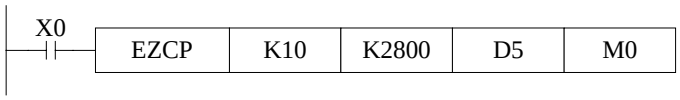
Function and action

Compare a float range with a float value



The status of the destination device will be kept even if the EZCP instruction is deactivated.

- The data of S1 is compared to the data of S2. The result is indicated by 3 bit devices specified with the head address entered as D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



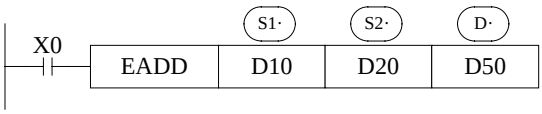
（K10）：[D6,D5]：（K2800）→M0，M1，M2

- Please set S1<S2, when S2>S1, see S2 as the same with S1 and compare them.

[EADD]

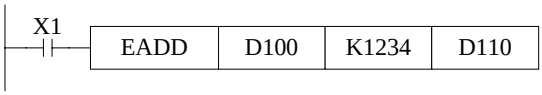
Operands：DX、DY、DM、DS、T、C、D、FD、K

Function



（D11,D10） + (D21,D20) → (D51,D50)

- The floating point values stored in the source devices S1 and S2 are algebraically added and the result stored in the destination device D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



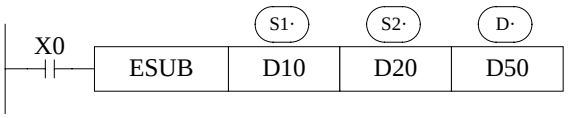
（K1234） + (D101,D100) → (D111,D110)

- The same device may be used as a source and as the destination. If this is the case then,on continuous operation of the EADD instruction, the result of the prevous operation will be used as a new source value and a new result calculated. This will happen every program scan unless the pulse modifier or an interlock program is used.

[ESUB]

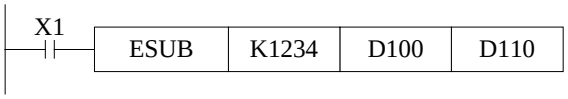
Operands：DX、DY、DM、DS、T、C、D、FD、K

Function and action



(D11,D10) － (D21,D20) → (D51,D50)

- The floating point value of S2 is subtracted from the floating point value of S1 and the result stored in destination device D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



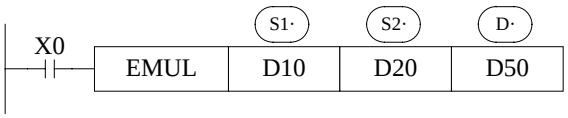
（K1234） － (D101,D100) → (D111,D110)

- The same device may be used as a source and as the destination. If this is the case then, on continuous operation of the EADD instruction, the result of the previous operation will be used as a new source value and a new result calculated. This will happen every program scan unless the pulse modifier or an interlock program is used.

[EMUL]

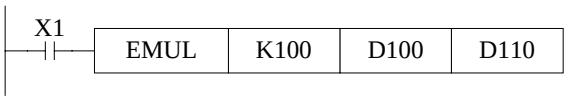
Operands：DX、DY、DM、DS、T、C、D、FD、K

Function and action



（D11，D10）×（D21,D20）→(D51,D50)

- The floating point value of S1 is multiplied with the floating point value point value of S2. The result of the multiplication is stored at D as a floating point value.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.

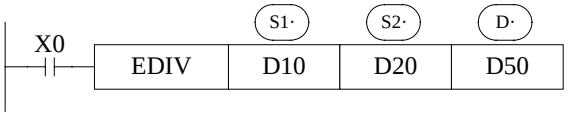


(K2346) × (D101,D100) → (111,D110)

[EDIV]

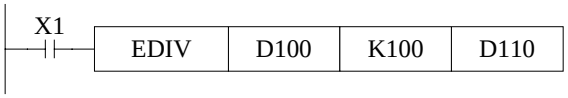
Operands：DX、DY、DM、DS、T、C、D、FD、K

Function



（D11,D10）÷（D21,D20）→（D51,D50）

- The floating point value of S1 is divided by the floating point value of S2. The result of the division is stored in D as a floating point value. No remainer is calculated.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



(D101,D100) ÷ (K2346) →(D111,D110)

- If S2 is zero then a divide by zero error occurs and the operation fails.

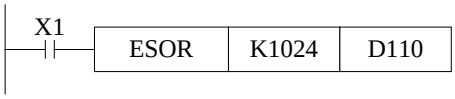
[ESQR]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function and action

(D11,D10) →（D21,D20）

- A square root is performed on the floating point value in S the result is stored in D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



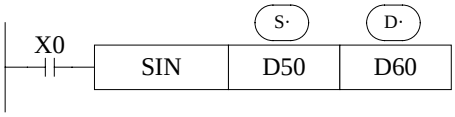
（K1024）→（D111，D110）

- When the result is zero, zero flag activates
- Only when the source data is positive will the operation be effective. If S is negative then an error occurs and error flag M8067 is set ON, the instruction can’t be executed.

[SIN]

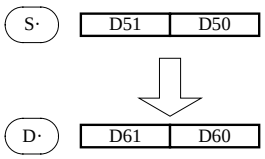
Operands：DX、DY、DM、DS、T、C、D、FD、K

Function



(D51,D50) → (D61,D60)SIN

- This instruction performs the mathematical SIN operation on the floating point value in S (angle RAD). The result is stored in D.



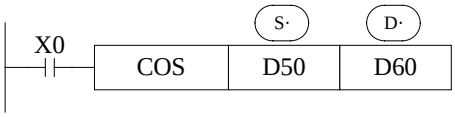
RAD value（angle×Π/180）
Assign binary float point value

SIN value
Binary float point

[COS]

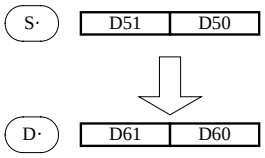
Operands：DX、DY、DM、DS、T、C、D、FD、K

Function and action



(D51,D50)RAD → (D61,D60)COS

- This instruction performs the mathematical COS operation on the floating point value in S (angle RAD). The result is stored in D.



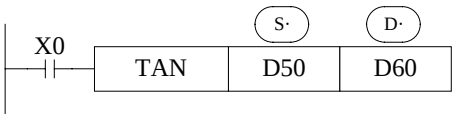
RAD value（angle×Π/180）
Assign binary float point value

COS value
Binary float point

[TAN]

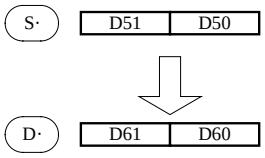
Operands：DX、DY、DM、DS、T、C、D、FD、K

Function



(D51,D50)RAD → (D61,D60)TAN

- This instruction performs the mathematical TAN operation on the floating point value in S. The result is stored in D.



RAD value（Angle×Π/180）
Assign binary float point value

TAN value
Binary float point

5-9. Clock operation

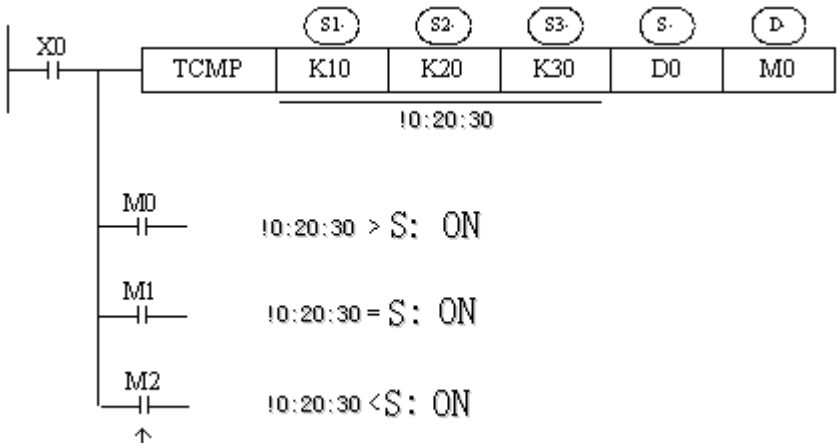
Mnemonic	Function
TCMP	Time Compare
TZCP	Time Zone Compare
TADD	Time Add
TSUB	Time Subtract
TRD	Read RTC data
TWR	Set RTC data

[TCMP]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function

Compare the assigned time with time data.



The status of the destination devices is kept, even if the TCMP instruction is deactivated.

- (S1·), (S2·) and (S3·) represent hours, minutes and seconds respectively. This time is compared to the time value in the 3 data devices specified by the head address (S·). The result is indicated in the 3 bit devices specified by the head address (D·).

(S1·) ： Hour
(S2·) ： Minute
(S3·) ： Second

(S·) ： Hour
(S·) ＋1 ： Minute
(S·) ＋2 ： Second

(D·)，(D·)＋1，(D·)＋2 ：According to the compare result, the 3 devices output ON/OFF.

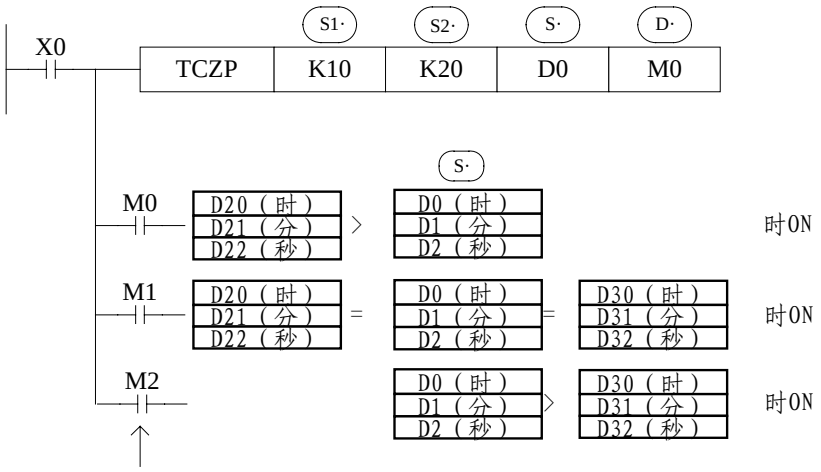
The valid range of “Hour” is「0~23」。
The valid range of “MInute” is「0~59」。
The valid range of “Second” is「0~59」。

[TZCP]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function and action

Compare the two assigned time with time data



The status of the destination devices is kept, even if the TZCP instruction is deactivated.

- (S1·), (S2·) and (S·) represent time values. Each specifying the head address of 3 data devices. (S·) is compared to the time period defined by (S1·) and (S2·). The result is indicated in the 3 bit devices specified by the head address (D·).

(S1·), (S1·)+1, (S1·)+2 : Assign the compare time's lower limit with the format of "Hour", "Minute" and "Second".

(S2·), (S2·)+1, (S2·)+2: Assign the compare time's lower limit with the format of "Hour", "Minute" and "Second".

(S·), (S·)+1, (S·)+2 : Assign the time data with the format of "Hour", "Minute" and "Second".

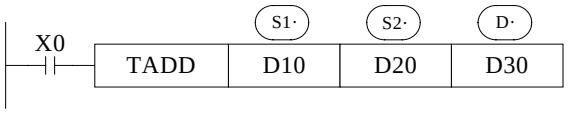
(D·), (D·)+1, (D·)+2 : According to the compare result, the 3 devices output ON/OFF.

The valid range of "Hour" is「0~23」。
The valid range of "Minute" is「0~59」。
The valid range of "Second" is「0~59」。

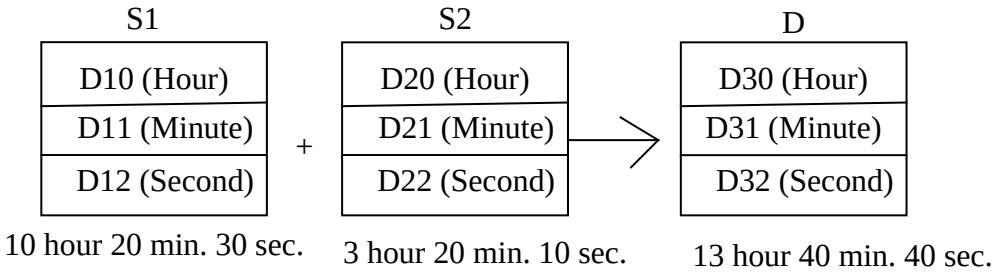
[TADD]

Operands：DX、DY、DM、DS、T、C、D、FD、K

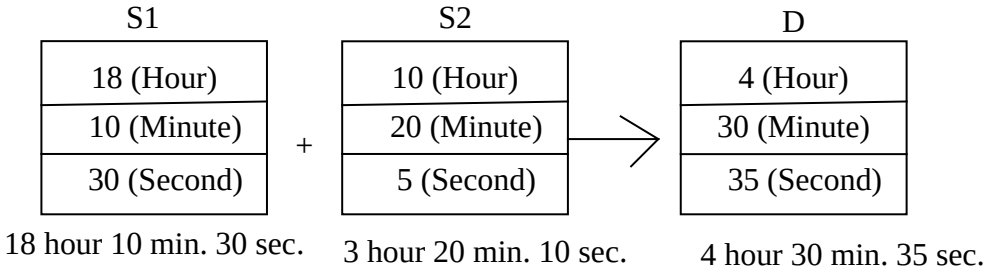
Function and action



(D10，D11，D12)＋（D20，D21，D22）→（D30，D31，D32）



- Each of S1, S2 and D specify the head address of 3 data devices to be used a time value. The time value in S1 is added to the value in S2, the result is stored to D as a new time value.
- If the addition of the two times results in a value greater than 24 hours, the value of the result is the time remaining above 24 hours. When this happens the carry flag M8022 is set ON.



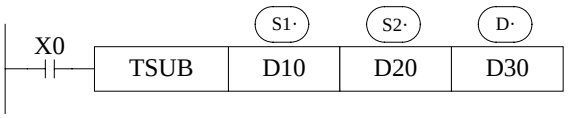
- When the result is 0（0 Hour 0 Minute 0 Second），Set zero flag ON.

The valid range of “Hour” is「0~23」。
The valid range of “Minute” is「0~59」。
The valid range of “Second” is「0~59」。

[TSUB]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function and action



(D10，D11，D12)－（D20，D21，D22）→（D30，D31，D32）

S1		S2		D
D10 (Hour)		D10 (Hour)		D10 (Hour)
D11 (Minute)	–	D11 (Minute)	=	D11 (Minute)
D12 (Second)		D12 (Second)		D12 (Second)
10 hour 20 min. 30 sec.		3 hour 20 min. 10 sec.		7 hour 0 min. 20 sec.

- Each of S1, S2 and D specify the head address of 3 data devices to be used a time value. The time value in S1 is subtracted from the time value in S2, the result is stored to D as a new time.
- If the subtraction of the two times results in a value less than 00:00:00 hours, the value of the result is the time remaining below 00:00:00 hours. When this happens the borrow flag M8021 is set ON.

S1		S2		D
10 (Hour)		18 (Hour)		4 (Hour)
20 (Minute)	–	10 (Minute)	=	30 (Minute)
5 (Second)		30 (Second)		35 (Second)
10 hour 20 min. 5 sec.		18 hour 10 min. 30 sec.		4 hour 30 min. 35 sec.

- When the result is 0 (0 hour 0 min. 0 sec.), zero flag set ON.

The valid range of “Hour” is「0~23」。
The valid range of “Minute” is「0~59」。
The valid range of “Second” is「0~59」。

[TRD]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function

The current time and date of the real time clock are read and stored in the 7 data devices specified by the head address D.

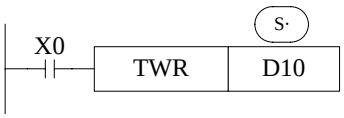
- Read PLC's real time clock according to the following format.
  The reading source is the special data register （D8013~D8019）which save clock data.

Device	Meaning	Values	Device	Meaning
D8018	Year	1~99	D0	Year
D8017	Month	1~12	D1	Month
D8016	Date	1~31	D2	Date
D8015	Hours	0~23	D3	Hours
D8014	Minutes	0~59	D4	Minutes
D8013	Seconds	0~59	D5	Seconds
D8019	Day	0 (Sat.)~6 (Sun.)	D6	Day

[TWR]

Operands：DX、DY、DM、DS、T、C、D、FD、K

Function



The 7 data devices specified with the head address S are used to set a new current value of the real time clock.

- Write the set clock data into PLC's real time clock.
  In order to wirte real time clock, the 7 data devices specified with the head address S should be set.

Device	Meaning	Values	Device	Meaning
D0	Year	1~99	D8018	Year
D1	Month	1~12	D8017	Month
D2	Date	1~31	D8016	Date
D3	Hours	0~23	D8015	Hours
D4	Minutes	0~59	D8014	Minutes
D5	Seconds	0~59	D8013	Seconds
D6	Day	0 (Sat.)~6 (Sun.)	D8019	Day

This instruction removes the need to use M8015 during real time clock setting. When setting the time it is a good idea to set the source data to a time a number of minutes ahead and then drive the instruction when the real time reaches this value.

MEMO

6. Special function instructions

In this chapter, we introduce the functions of high speed count input, high speed output and MODBUS communication instructions of XC series PLC.

6-1. High speed count

6-1-1. Interior high speed counter's No. and function

High speed counter's --

Interior high speed counter's No. is in the following table. They're allocated in the input X000~X005 according to the counter's No. that cannot be used repeatedly.

When X000~X005 don't used as high speed count input, they could be used as normal input points.

[U]: count pulse input;

[D]: count direction judgment (OFF is +, ON is -);

[A]: A phase input;

[B]: B phase input

	Single phase count										Single phase+/- input count					AB phase count		
	C600	C602	C604	C606	C608	C610	C612	C614	C616	C618	C620	C622	C624	C626	C628	C630	C632	C634
X000	U										U					A		
X001		U									D					B		
X002																		
X003			U									U					A	
X004												D					B	
X005																		
X006				U									U					A
X007													D					B

Function

High speed counter executes according to the format in the upward table and to the special inputs. Go on high speed action according to the interrupt disposal. It's independent with the PLC's scan cycle.

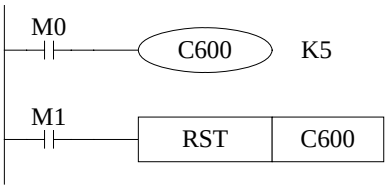
This type of counters could be divided into the following three types:

Item	Single phase positive count input	Single phase +/- count input	AB phase count input
Count direction's assign method	Only positive count is ok	When direction judgment input is OFF, pulse input is positive count; When direction judgment input is ON, pulse input is negative count	A phase exceed B phase 90 ^{o} positive count; A phase lag B phase 90^{o} negative count;

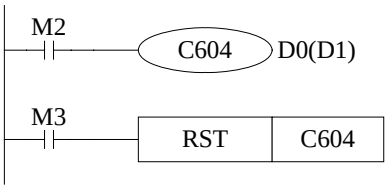
6-1-2. Using method of single phase high speed counter

Action

Single phase positive count

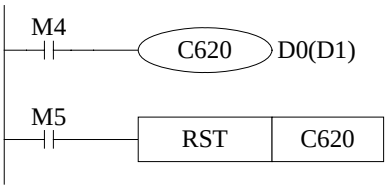


- When M0 is ON，C600 count with OFF→ON of X000
- If M1 is ON, reset when execute RST instruction

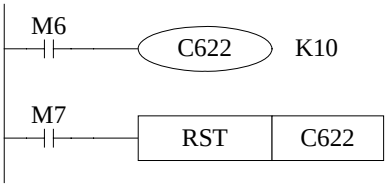


- C604 starts to count when M2 is ON. The count input is X004, in this examples, the set value use the content in the indirect data register.
- As showed in the above graph, execute reset via M3 in the program.

Single phase positive/negative count



- When M4 is ON, C620 counts with OFF→ON of X000. Via OFF or ON of X001, judge the count direction. If X001 is OFF，execute positive count, if X001 is ON, execute negative count.

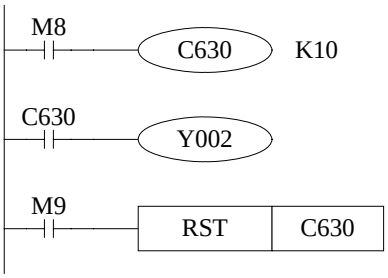


- When M2 is ON, C622 counts with OFF→ON of X000. Via OFF or ON of X002, judge the direction. If X003 is OFF, execute positive count, if X003 is ON, execute negative count.

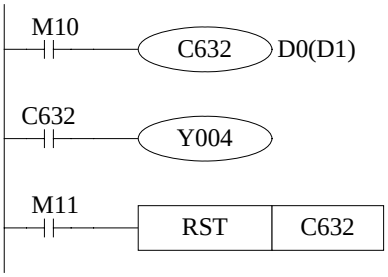
6-1-3.Using method of AB phase high speed counter

AB phase input

AB phase input counter executes increment/decrement count via the judgment of A、B phase. The output contact's correspond with the current value is the same with the preceding single high speed counter.



- When M8 is ON, C630 counts with the input X000(A phase)、X001(B phase) via interrupt.
  If M9 is ON, execute RST.
- If the current value exceeds the set value, then Y002 is ON；if the current value is less than the set value, then be OFF.

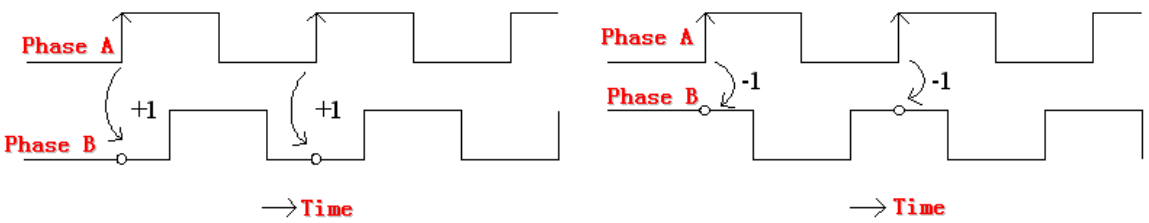


- When M10 is ON, C632 starts to count immediately. Counter's input is X002(A phase)、X003(B phase).
- Reset in the sequencial control program via M11.
- If the current value is larger than the pre-set value, Y004 activates, if lower than the pre-set value, then cut.

- In the case of A phase input is OFF→ON, at the same time B phase input is OFF, the counter carries on increment count. If at the same time B phase input is ON, the counter carries on decrement count.

Dual phase input

- The normal rotate encoder's output is phase A and phase B which have 90° phase difference. According to this ,high speed counter automatic execute increment/decrement count.
- This dual phase counter activates as counter which increases one

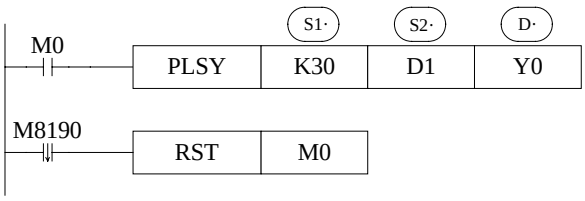


6-2. Pulse output

6-2-1.Pulse output [PLSY]

Operands：Y、DX、DY、DM、DS、T、C、D、K

Function and Action



- The instruction with the assigned frequency to generate the assigned pulse；support 32 bits instruction [DPLSY].

S1: assign the frequency. The bound: 0~400KHz

S2: assign the generated pulse quantity.

The allowed setting bound: 16 bits instruction→0~32,767

32 bits instruction→0~2,147,483,647

when the pulse setting number is 0, don't send pulse; assign the value as H 7FFFFFFF, there is no limitation with the generated pulse number.

D: assign output pulse's Y number, can only output at Y000 or Y001

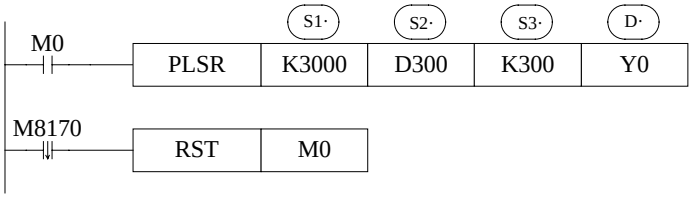
- When M0 is ON, with this PLSY instruction, Y0 output pulses with the output frequency 30Hz, the pulse number is assigned by D1, if set pulse number as H 7FFFFFFF, it means send infinity pulses , at this time coil M8190 set ON. When the output pulse number reaches the set value, stop outputting the pulse, at this time, coil M8190 set OFF, reset M0.

6-2-2. [PLSR] with speedup/speed-down pulse

Operands：Y、DX、DY、DM、DS、T、C、D、K

Function

1、pulse output of single segment and single direction



- The instruction with the assigned frequency to generate a certain quantity pulse; support 32 bit instruction [DPLSR].

  "S1": highest frequency. The bound is：0~400KHz

  "S2": total output pulse number.

  Allowed setting bound：16 bits instruction→0~32,767

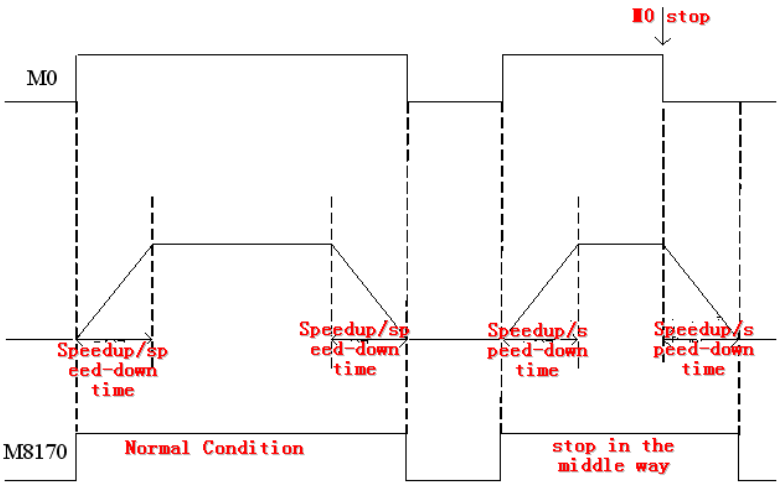
  32 bits instruction→0~2,147,483,647

  If set the pulse number as 0, don't send pulse; if assign this value as H 7FFFFFFF, there will be no limitation for the generate pulse number.

  "S3": speedup/speed-down time. The set bound：below 5000ms

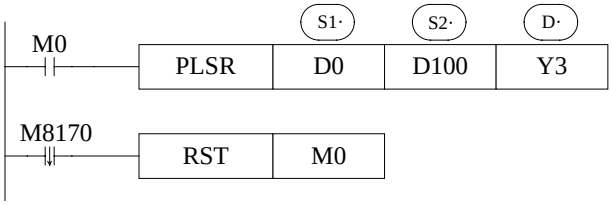
  "D": assign Y number of output pulse, could only be output at Y000 or Y001

- When M0 is ON, PLSR starts pulse output, send assigned pulse number according to the assigned speedup/speed-down slope、highest frequency. To output with the constant speed, set the speedup/speed-down time as 0. If set the pulse number as H 7FFFFFFF, infinity pulse number will be sold out, at this time coil M8170 set ON.
- When the output pulse number reaches the set value, stop pulse outputting, at this time coil M8170 set OFF, reset M000. See the following chart
- If pulse output M000 is OFF, pulse output decreases to be 0 according to the assigned slope. Stop pulse outputting, coil M8170 set OFF.

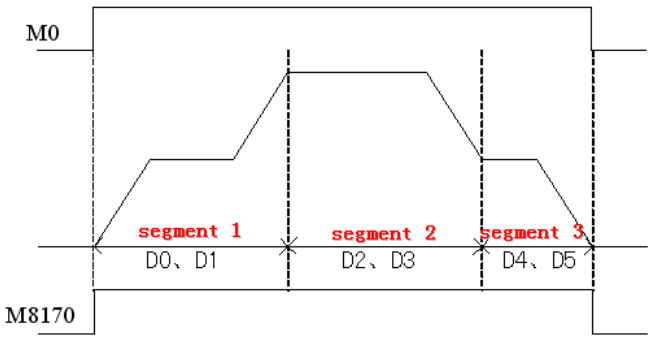


Function

2、Pulse output of segments and single phase

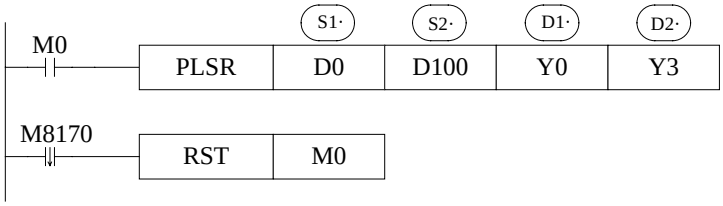


- The instruction which generates a certain quantity pulse with the assigned frequency.
  S1: an area with Dn or FDn as the start address. In the above example, D0 set the highest frequency of segment 1 pulse, D1 set the highest frequency of segment 1 pulse, D2 set the highest frequency of segment 2 pulse, D3 set the highest frequency of segment 2 pulse, ……if the set value of Dn, Dn+1 are both 0, it means segment finish.
  S2: speedup/speed-down time. Here the time means the speed time from start to the first segment's speedup time, meantime, all segments' frequency and time slope are defined. So the following speedup/speed-down speed follows them. The set bound is: Below 5000ms.
  D: assign the Y number of output pulse, can only output at Y000 or Y001
- Support double words output DPLSR, here D0、D1 set the highest frequency of segment 1、D2、D3 set the pulse number of segment 1, D4、D5 set the highest frequency of segment 2、D6、D7 set the pulse number of segment 2……



Function

3、pulse output of segment dual direction



- Instruction of generate a certain quantity pulse with the assigned frequency.

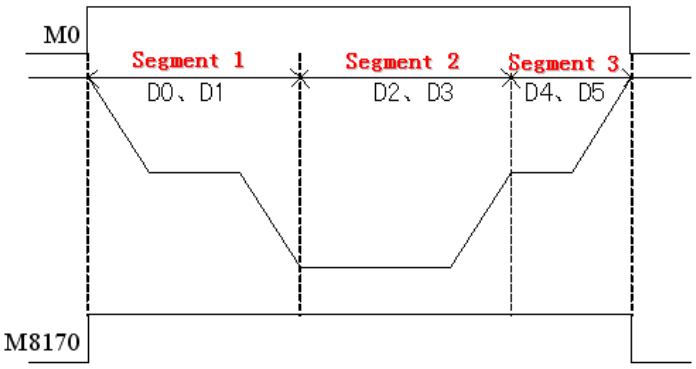
  "S1" : an area take Dn or FDn with the start address. In the preceding example, D0 set the max frequency of segment 1, D1 set pulse number of segment 1. D2 set the max frequency of segment 2, D3 set pulse number of segment 2, …… if Dn、Dn+1 are both 0, it means segment finish.

  "S2": speedup/speed-down time, here the time means the speedup time from the start to the highest frequency. At the same time all segments' frequency and time slope is defined, so the following speedup/speed-down format all do according to them. The set bound: below 5000ms

  "D1": assign Y number of output pulse, can only output at Y000 or Y001

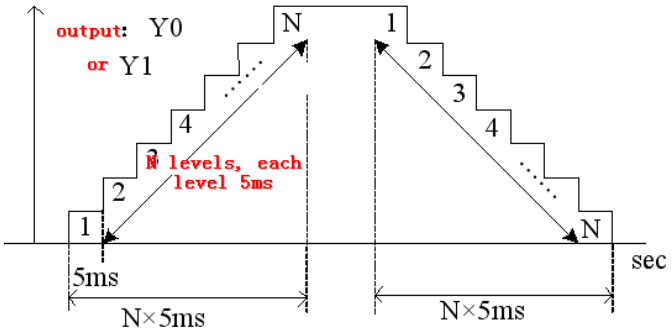
  "D2": assign Y number of output pulse direction, can be assigned at your will. E.g. In "S1", if the pulse number is a positive value in segment 1, Y output ON; if be negative, Y is OFF.

Please note: in once segment pulse output, pulse's direction is only determined by the pulse number set value (positive or negative) of the first segment.



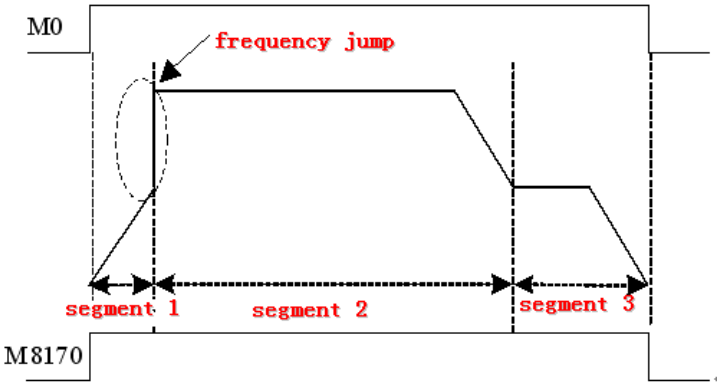
Note Items

1、some basic parameters of speedup/speed-down



- In the process of speedup/speed-down, each step's time is 5ms, this time is fixed.
- The max. step is 15K. (the increase/decrease frequency of each step). If the value exceeds 15K, count as 15K； the minimum step frequency is 10Hz, if lower than 10Hz, calculate as 10Hz.
- When carrying on pulse output, please note each segment's pulse number shouldn't lower than 10, if the set value is less than 10, sent as 10.

2、Frequency jump in segment pulse output

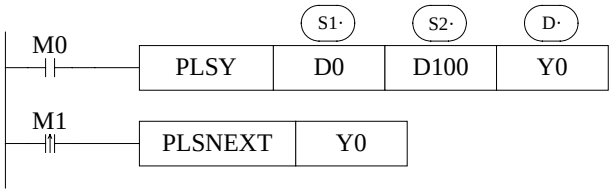


- In the process of segment pulse output, if the current pulse number has sent out but still haven't reached the current segment's max. frequency, then in the process from the current segment to the next pulse output, there will be pulse frequency jump. See the following chart.
- To avoid frequency jump, please note the speedup/speed-down time set value not to small.

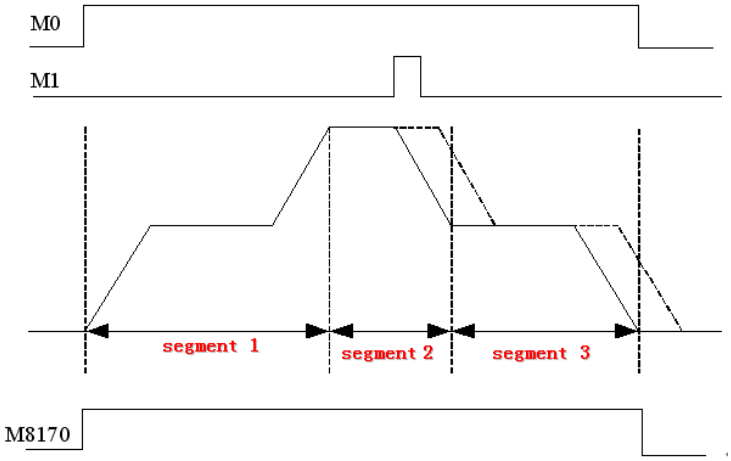
6-2-3. [PLSNEXT] pulse segment shift

Operands：Y、DX、DY、DM、DS、T、C、D、K

Function and Action



- In the condition of pulse output reaches the highest current value, then output stable under this frequency, if M1turns from OFF to ON, enter the next pulse output with the speedup/speed-down time
- Please note, in the process of pulse's speedup/speed-download, executing this instruction is invalid.

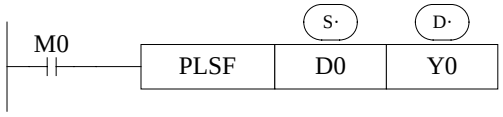


--------（the broken line）means the original pulse output curve

6-2-4. [PLSF] for alterable frequency output

Operands：Y、DX、DY、DM、DS、T、C、D、K

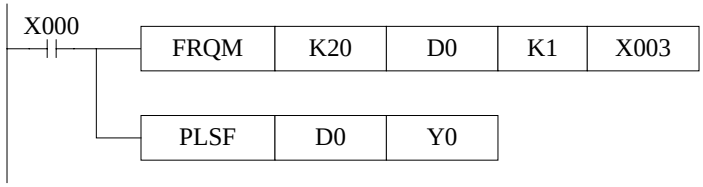
Function and Action



- Generate the sequencial pulse via change frequency's format；
- Support 32 bits instruction [DPLSF]。
- "S" assign pulse's frequency, the bound：200~400KHz
  When set the frequency lower than 200Hz, output the frequency with 200Hz
- "D" assign Y number of pulse output, can only output at Y000 or Y001.
- In the above example, with the changing of setting frequency in D0, the frequency output from Y0 changes.

Example

The following example, the pulse output frequency of Y0 equals the input frequency from X003. When the frequency from X003 changes, Y0 output frequency changes together!

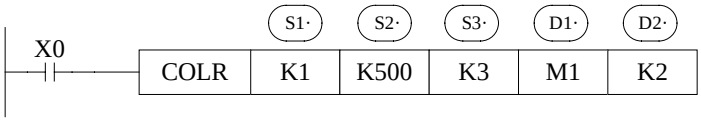


6-3. MODBUS communication instruction

Operands：DX、DY、DM、DS、T、C、D、K

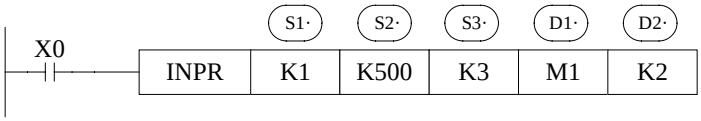
Function and action

1、Coil Read [COLR]



- The command read the assigned bureau's assigned coil to the assigned coil.
  - (S1·) Communication bureau
  - (S2·) Coil's start ID
  - (S3·) Coil's number
  - (D1·) Receive coil's start address
  - (D2·) Serial port's ID. Bound:1~3

2、Input coil read [INPR]

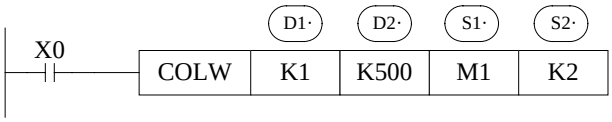


- Read the assigned bureau's assigned input coil to the model's assigned coil.
  - (S1·) Communication bureau ID
  - (S2·) Coil's start ID
  - (S3·) Coil's number
  - (D1·) Receive coil's start address
  - (D2·) Serial port's ID. Bound: 1~3

Instruction description: When X0 is ON, execute COLR or INPR instruction. After finish executing the instruction, set the communication finish flag. When X0 is OFF, no operation. When communication error occurs, repost automatically. When reach 10 times, set communication error flag. User could find the reason why the correspond register judge error.

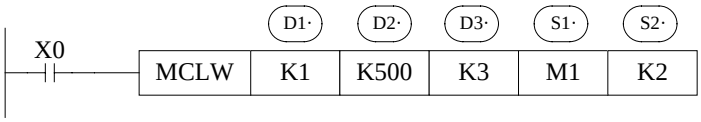
Function and action

3、Single coil write [COLW]



- Write the model's assigned coil to the assigned bureau's assigned coil
  - (D1·) Communication bureau ID
  - (D2·) Coil's start ID
  - (S1·) Receive coil's start address
  - (S2·) Serial port's ID. Bound: 1~3

4、Multi-coil write [MCLW]

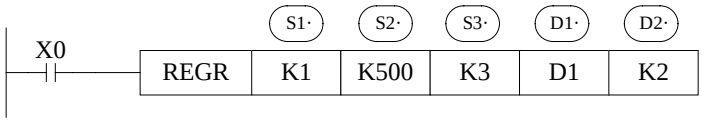


- Write the model's assigned multi-coil to the assigned bureau's assigned coil
  - (D1·) Communication bureau ID
  - (D2·) Coil's start ID
  - (D3·) Coil's number
  - (S1·) Receive coil's start coil address
  - (S2·) Serial port's ID. Bound: 1~3

Instruction description: When X0 is ON, execute COLW or MCLW instruction. When finish executing the instruction, set communication finish flag. When X0 is OFF, no operation. If communication error, repost automatically. When reach ten times, set communication error flag. User could inquiry about the reason why related register judge error.

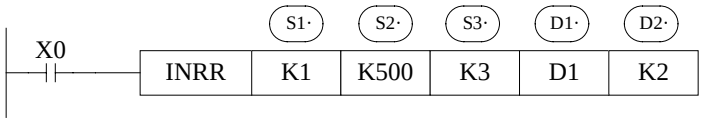
Function and action

5、Register read [REGR]



- Read the model's assigned register to the assigned bureau's assigned register
  - (S1·) Communication bureau ID
  - (S2·) Register's start ID
  - (S3·) Register's number
  - (D1·) Receive register's start address
  - (D2·) Serial port's ID. Bound: 1~3

6、Input register read [INRR]

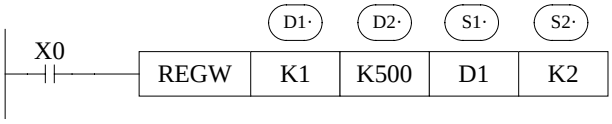


- Read the model's assigned input register to the assigned bureau's assigned register 。
  - (S1·) Communication bureau ID
  - (S2·) Register's start ID
  - (S3·) Register's number
  - (D1·) Receive register's start ID
  - (D2·) Serial port's ID. Bound: 1~3

Instruction description：When X0 is ON, execute REGR or INRR instruction. When finish executing, set communication finish flag. When X0 is OFF, no operation. If communication error, repost automatically. When reach 10 times, set communication error flag. User could inquiry the reason why related register judge error.

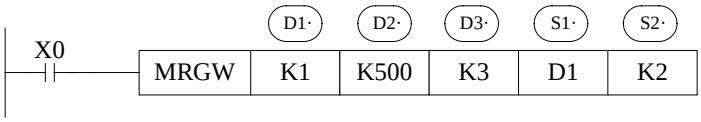
Function and action

7、Single register write [REGW]



- Read the model's assigned register to the assigned bureau's assigned register.
  - (D1·) Communication's bureau ID
  - (D2·) Register's start ID
  - (S1·) Receive register's start ID
  - (S2·) Serial port's ID. Bound: 1~3

8、Multi-register write [MRGW]



- Read the model's assigned input register to the assigned bureau's assigned register.
  - (S1·) Communication bureau ID
  - (S2·) Register's start ID
  - (S3·) Register's number
  - (D1·) Receive register's start address
  - (D2·) Serial port's ID. Bound: 1~3

Instruction description: When X0 is ON, execute REGW or MRGW instruction. After finish executing the instruction, set the communication finish flag. When X0 is OFF, no operation. If communication error, repost automatically. When reach 10 times, set communication error flag. User could inquiry the reason why related register judge error.

6-4. Free format communication

Description of COM ports

XC series PLC has 3 communication ports, free format communication usually use COM1 or COM2. COM1 is RS-232 port, COM2 is RS-232 port or RS-485 port.

Via COM1、COM2, communicate with host machine、display、other serial communication device. Communication protocol has program protocol and free format etc.

Description of free format communication

Free Communication

Free format communication transfer data in the format of data block, each block transfer 128 bytes at most! At the same time, each block could set a start symbol and an end symbol, or you needn't set.

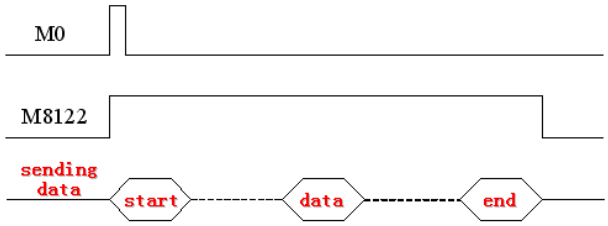
Start symbol (1 byte)	Data block (Max.128 bytes)	End symbol (1 byte)

- Data format
  Data bit：7bits、8bits
  Checkout bit：odd check、even check、no check
  Stop bit：1 bit、2 bit
- Start symbol：1 bit
  End symbol：1 bit
  The user could set a start/end symbol, after setting the start/end symbol, when PLC sending data, start/end symbol will automatic be added. When receiving data, automatic delete the start/end symbol.
- Communication format：8 bits、16 bits
  When choose 8 bits cushion format to communicate, the register's high byte is invalid in the communication process, PLC only use register's low bytes to send and receive data.

Format

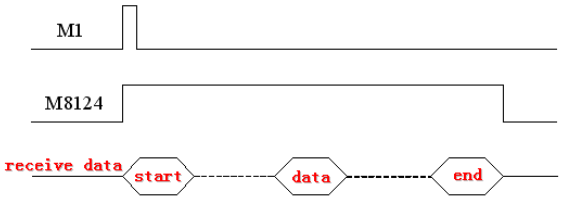
1、sending data：

- Data sending instruction, once rising edge of M0 send once data.
  (S1·) Send data's first ID.
  (S2·) ASC number sent
  n：COM port
- In the process of data sending, "sending" flag M8122 (COM1) set ON.



2、receiving data：

- Data receive instruction, the rising edge of M0 receive the data.
  (S1·) Receive data's start ID.
  (S2·) The max ASC number received
  n： COM port
- In the process of data receiving, "receiving" flag M8124 (COM1) set ON.



Parameter Setting

About XC series free communication, COM1、COM2 are both available. To the two COM ports, communication parameters should be set separately.

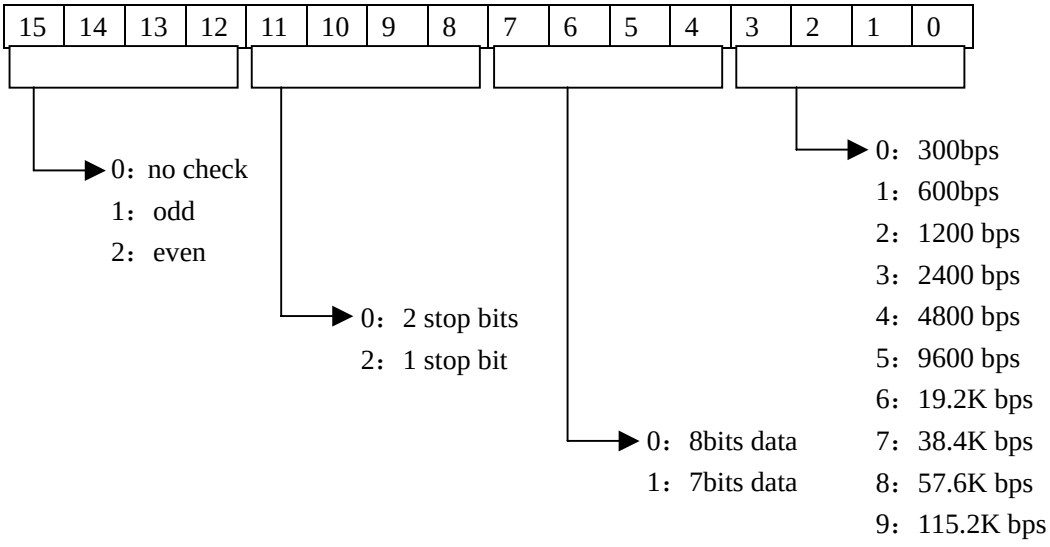
for communication parameter's setting, please refer to the following table：

	Number	Function	Description
COM1	FD8210	Communication mode	255 is free format 1~254 bits modbus station ID
	FD8211	Communication format	Baud rate, data bit, stop bit, checkout
	FD8212	ASC timeout judgment time	Unit: ms, if set to be 0, it means no timeout waiting
	FD8213	Reply timeout judgment time	Unit: ms, if set to be 0, it means no timeout waiting
	FD8214	Start symbol	The high 8 bits are invalid
	FD8215	End symbol	The high 8 bits are invalid
	FD8216	Free format setting	8/16 bits cushion， have/no start bit have/no end bit

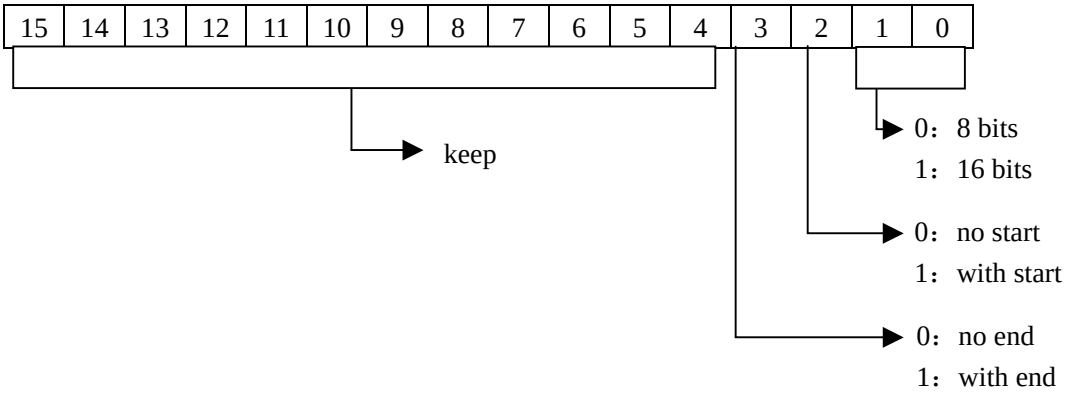
COM2	FD8220	Communication mode	255 is free format 1~254 bits modbus station ID
	FD8221	Communication format	Baud rate, data bit, stop bit, checkout
	FD8222	ASC timeout judgment time	Unit: ms, if set to be 0, it means no timeout waiting
	FD8223	Reply timeout judgment time	Unit: ms, if set to be 0, it means no timeout waiting
	FD8224	Start symbol	The high 8 bits are invalid
	FD8225	End symbol	The high 8 bits are invalid
	FD8226	Free format setting	8/16 bits cushion， have/no start bit have/no end bit

Setting method of communication parameter:

FD8210（COM1）/FD8220（COM2）：

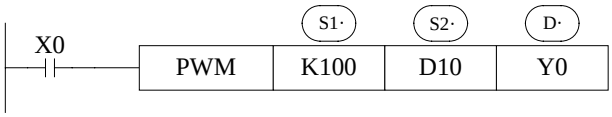


FD8216（COM1）/FD8226（COM2）：



6-5．PWM pulse width modify

Function and action



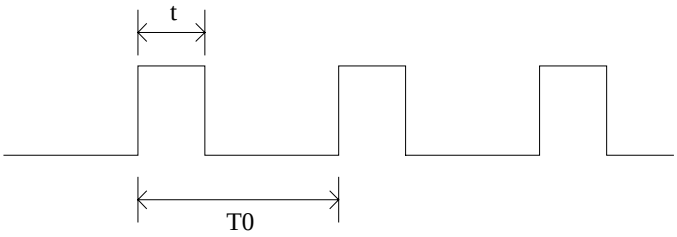
(S1·) Assign occupy/empty ratio value “n”. The bound is：1~255

(S2·) Assign output frequency f. The bound is：0~72KHz

(D·) Assign Y number of output pulse

Can only output at Y000 or Y001 (please treat as transistor output type).

- The output occupy/empty ratio of PMW =n /256×100%
- PWM output use the unit of 0.1Hz, so when set (S1) frequency, the set value is 10 times of the actual frequency (i.e. 10f). E.g.：to set the frequency as 72KHz, then set value in (S1) as 720000.
- When X000 is ON, output PWM wave；when X000 is OFF, stop outputting. PMW output doesn’t have pulse accumulation.

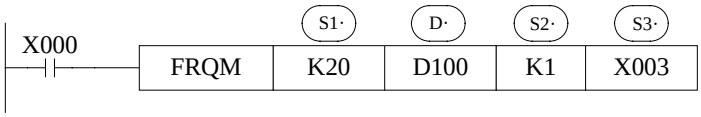


In the upward graph：T0=1/f

T/T0=n/256

6-6. Frequency testing

Function and Action



(S1) pulse cycle number. I.e. In one scan cycle, collect tested input pulse cycle number.

(S2) testing result. Operands：D、CD、TD

Frequency choosing. Choose bound：K1 or K2；

When frequency is K1, frequency testing bound：>= 9Hz, precise bound：9~18KHz.
When frequency is K2, frequency testing bound：>=300Hz，precise bound：300~400KHz。

Pulse input port X number.

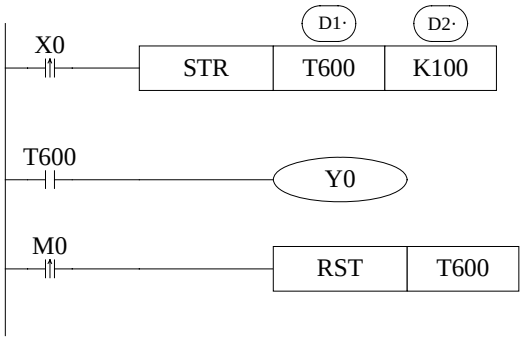
- When testing the frequency, choose the frequency as K2, the frequency testing precise is higher than K1.
- When X000 is ON, each scan cycle of FRQM test 20 pulse cycle from X003, calculate the frequency value and store into D100, repeatedly testing. If the tested value is smaller than the tested bound, return the tested value as 0.

Table of X number correspond with frequency testing pulse output

Model	X number
XC3- 14 model	X2、X3
XC3-24、XC3-32 model	X1、X11、X12
XC3-48、XC3-60 model	X4、X5
XC3-18R model	X1、X6、X7

6-7. Precise time

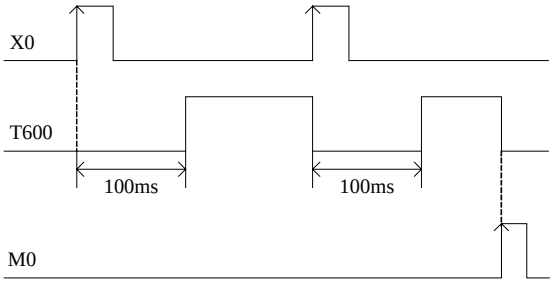
Function and Action



D1·: timer's number. The bound：T600~T618（T600、T602、T604…T618）

D2·: the time value. Operands: constant、register

- This instruction is the precise time instruction with the cycle of 1ms.
- When X000 turns from OFF to ON, timer T600 starts to time, when time accumulation reaches 100ms, T600 set； if X000 again turns from OFF to ON, timer T600 turns from ON to OFF， restart to time, when time accumulation reaches 100ms, T600 again reset. See the following chart.
- When time reaches, T600 activates, then execute the interrupt program with the interrupt tag I3001. for each timer's correspond interrupt tag, please refer to the following table:



Interrupt tag correspond with the timer：

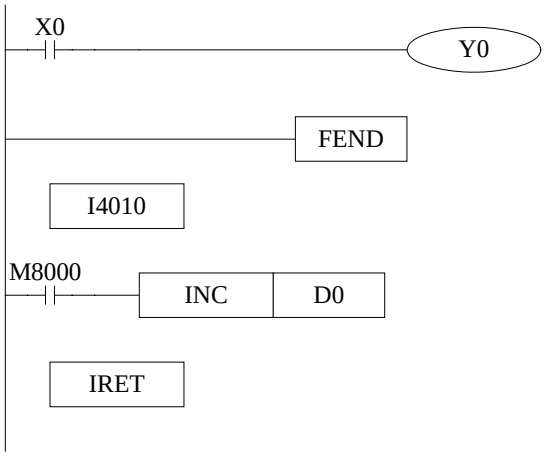
Timer's number	Interrupt tag
T600	I3001
T602	I3002
T604	I3003
T606	I3004
T608	I3005
T610	I3006
T612	I3007
T614	I3008
T616	I3004
T618	I3010

6-8．Interrupt function

6-8-1.Time interrupt

Function and action

In the condition of the main program has a long executing cycle, if you want to dispose special program, or in the sequencial control scan, a special program should be executed at every certain interval, time interrupt function should be used. It could be not effected by PLC's scan cycle, execute interrupt subroutine every Nms interval.



- Time interrupt subroutine is similar with other interrupt subroutines, they must be written behind the main program, start with the instruction I40xx, end with IRET.
- In I40xx,‘xx’means interrupt time, the unit is ms. E.g.：I4010 means every 10ms interval, execute once interrupt.
- Totally there are 10 routes time interruption, from I40xx~I49xx。

7．Applied example programs

In this chapter, we give you some sample programs for your reference.

XC series PLC is mini model、high speed、good performance PLC. Besides the independent using of I/O points, pulse output and other functions could be used. So XC series PLC could satisfy diverse control.

7-1. Example of pulse output

E.g：The following is the program which realize continuous sending high-low pulse

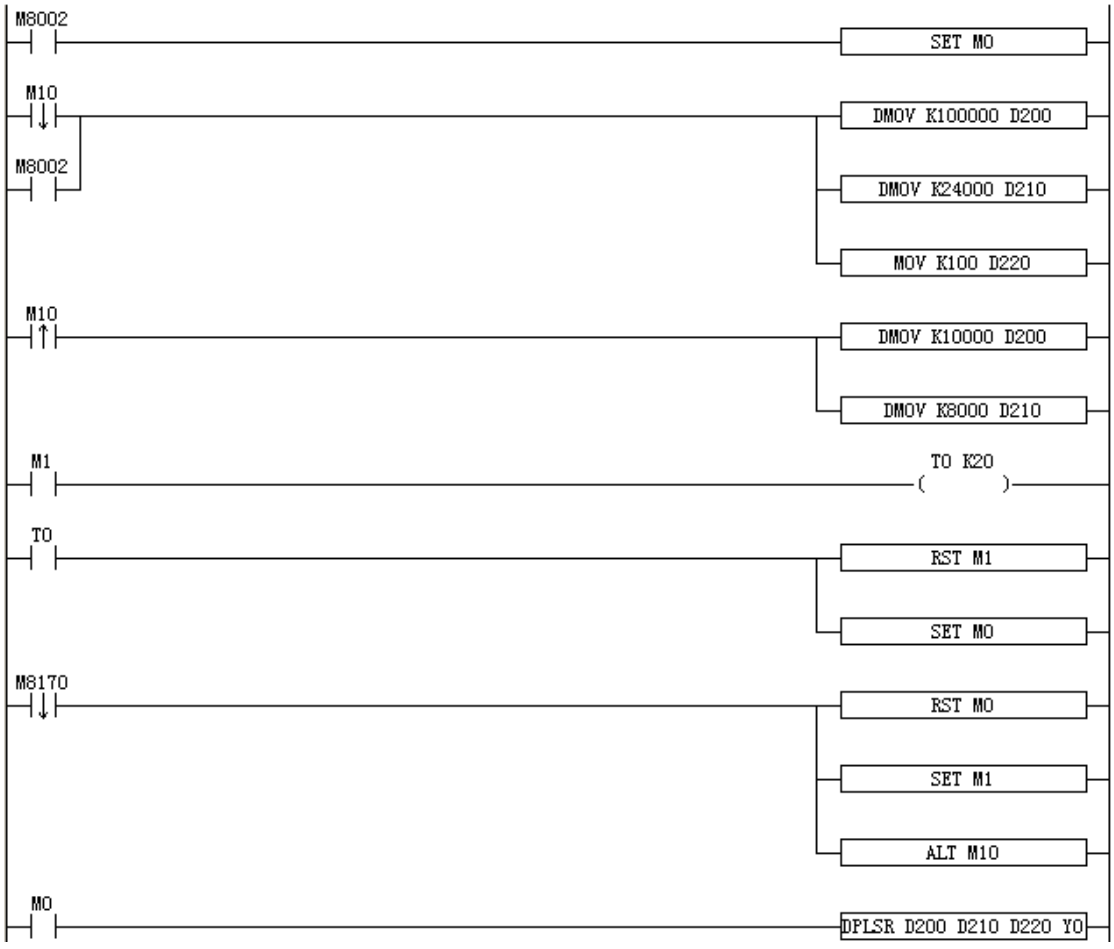
The parameters：

Parameters of step motor：step square angle =1.8 degree/step, fractionlet =40，the pulse number of a round is 8000。

High frequency pulse：max frequency is 100KHz，the total pulse number is 24000（3 rounds）

Low frequency pulse：Max frequency 10KHz，total pulse number is 8000（1 round）

Ladder program：



Program description：

When PLC changes from STOP to RUN, M8002 coil gets through a scan cycle, set high frequency pulse parameters into D200、D210, set speedup/speed-down time into D220, set M0, the motor start to speedup with high frequency and work 3 rounds, set coil M8170 at the same time; the motor runs 3 rounds, the speed-down till stop, coil M8170 reset. Here reset M0, set M1, reverse M10 status, set low frequency parameters into D200、D210. the counter starts to delay with 2 seconds, when reach this 2 seconds, M1 is reset, M0 is set again, the motor starts to run 1 round with low frequency. After finish this 1 round, the motor starts to run with high frequency again! In this format, the motor runs with high frequency and low frequency.

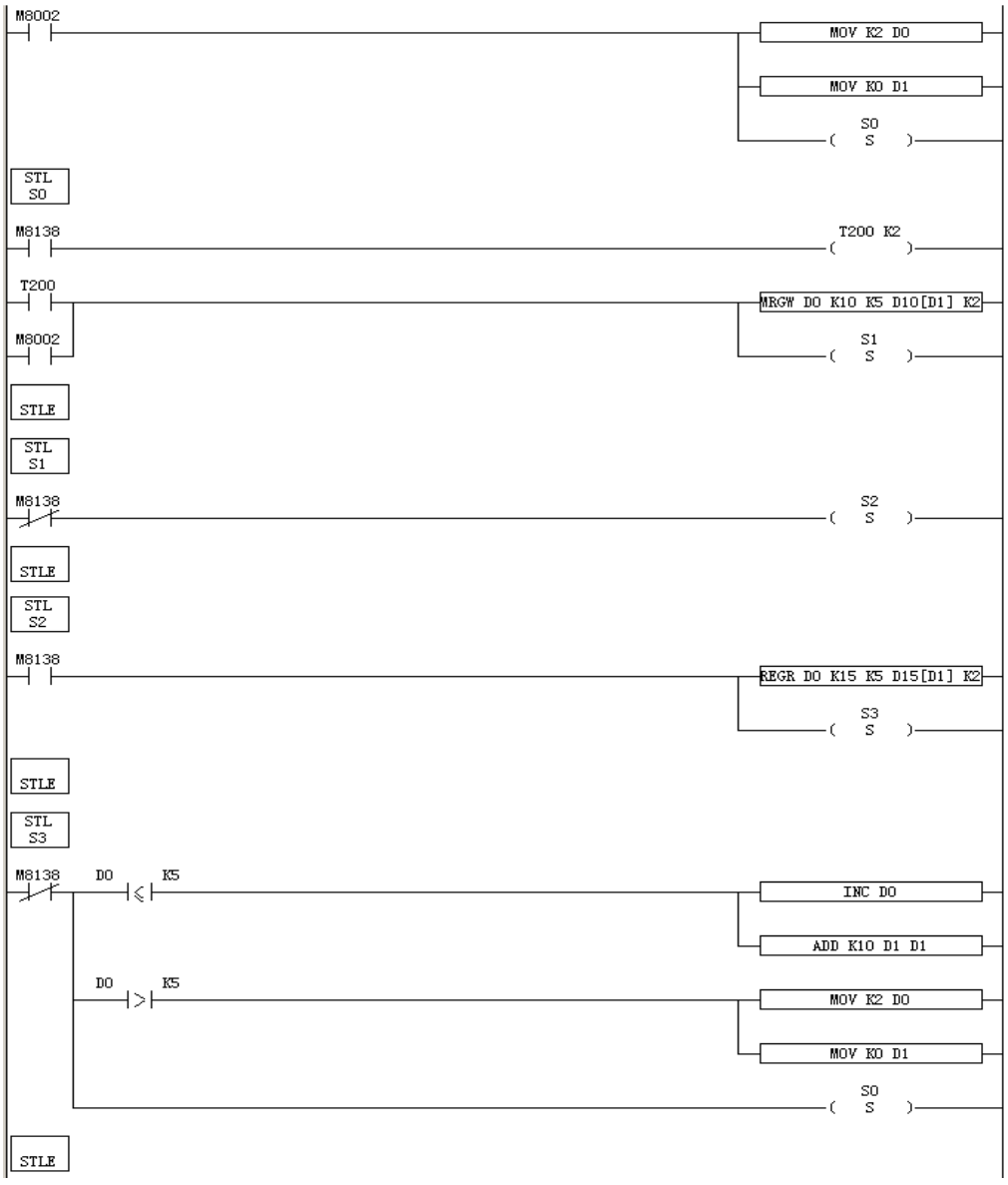
7-2. Example of MODBUS instructions

E.g.: The following is the communication program of one master station and 4 slave stations

Each parameters:

The master station number is 1, slave stations numbers are 2，3，4，5. This example, we use No.2 communication port:

Ladder program:



Program description：

When PLC changes from STOP to RUN, M8002 coil gets through a scan cycle. S0 flow open, write master machine's D10——D14 into No.2 slave machine's D10——D14. after finish communication, set M8138, at the same time write slave machine's D15——D19 into master machine's D15——D19，set communication finish flag. Realize write and read to a slave station. At this time flow S3 will judge with the slave station. If the station number is less than 5，station number add 1，offset add 10；or else station number starts from number 2 station again.

7-3．Example of free format communication

This example is the free format program with DH107/DH108 series instruments:

I、Interface specification

DH107/DH108 series instruments use asynchronism serial communication ports, the interface level fits the standard of RS232C or RS485. the data format is 1 start bit, 8 bits data, no check bit, one or two stop bits. Baud rate of communication transfer data could modified to be 1200~19200bit/s

II、Format of communication instructions

DH107/108 instruments use Hex. data format to indicate each instruction code and data.

Read/write instruction:

Read：The address code +52H（82）+parameter's (to read) code+0+0+CRC check code

Write：The address code +43H（67）+ parameter's (to write) code +the write data's low byte +the write data's high byte +CRC check code

Read instruction's CRC check code is：parameter's (To read) code *256+82+ADDR

ADDR is instrument's ID value, the bound is 0~100 (please do not add 80H). CRC is the redundant caused by the following operation: the preceding data operate with binary 16 bits integer plus. The redundant is 2 bytes, the low byte is ahead, the high byte is behind

Write instruction's CRC check code is：parameter's (to write) code *256+67+parameter's (to write) value +ADDR

The parameter's (to write) value is indicated by Hex. binary integer

No matter write or read, the instruments will return the following data

The test value PV+ the given value SV+ the output value MV and alarm status + read/written parameter's value +CRC check code

PV、SV and the read parameter's value should be integer format, each engrosses 2 bytes, MV engrosses one byte, the data bound is 0~220, the alarm status engrosses one byte, CRC check code engross 2 bytes, the total is 10 bytes.

CRC check code is PV+SV+（alarm status *256+MV）+parameter's value +ADDR，the redundant caused by the integer plus

(the detailed format, please refer to AIBUS communication protocol description) .

III、Compile communication program

After power on, the program read the current temperature value every 40ms. In this period the user could also write the set temperature value.

Data area definition：send data buffer area: D10~D19

Accept data buffer area: D20~D29

Instrument's station ID: D30

Read command's value: D31=52 H

Write command's value: D32=43 H

Parameter's code: D33

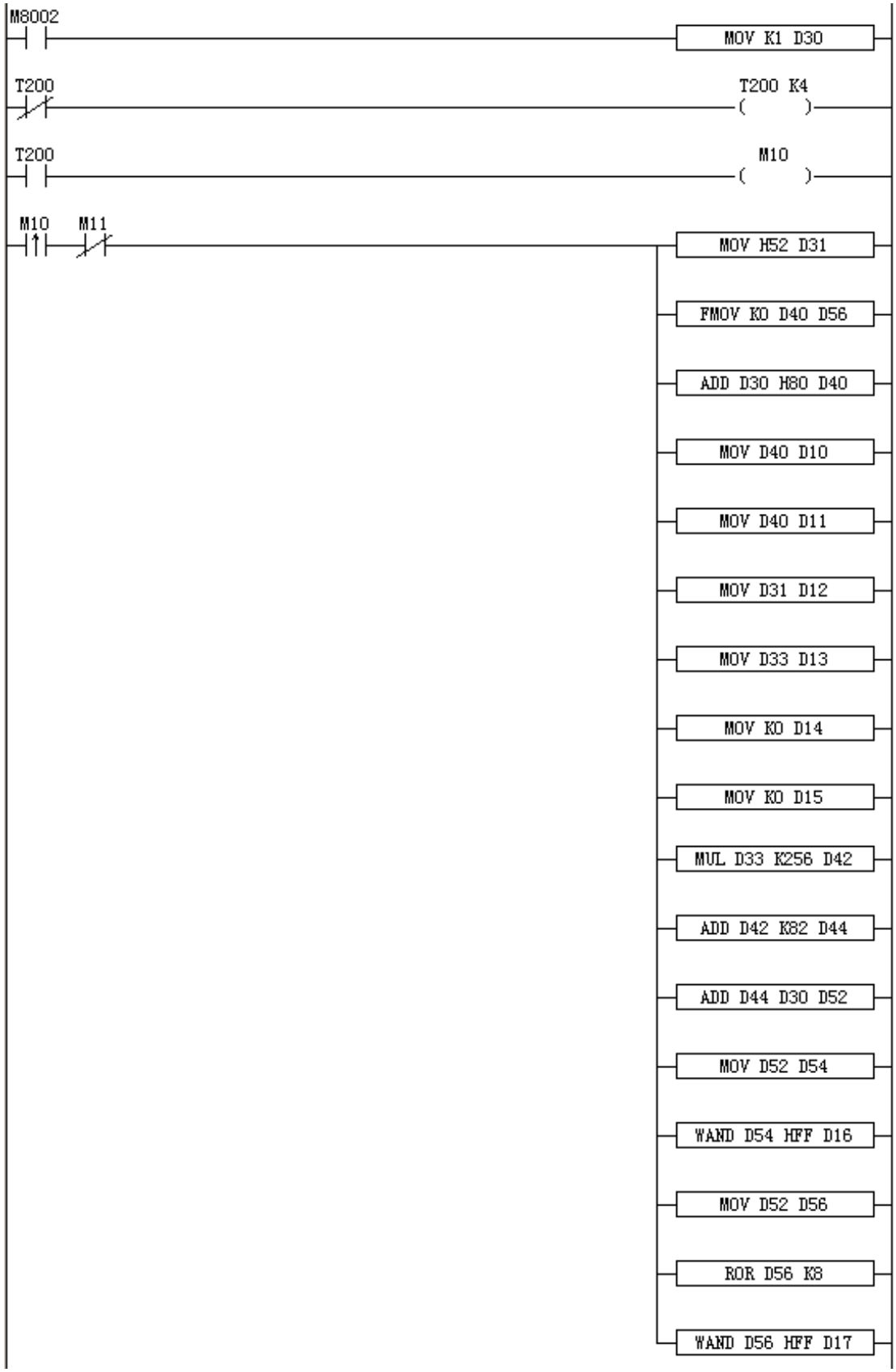
Temperature setting: D34

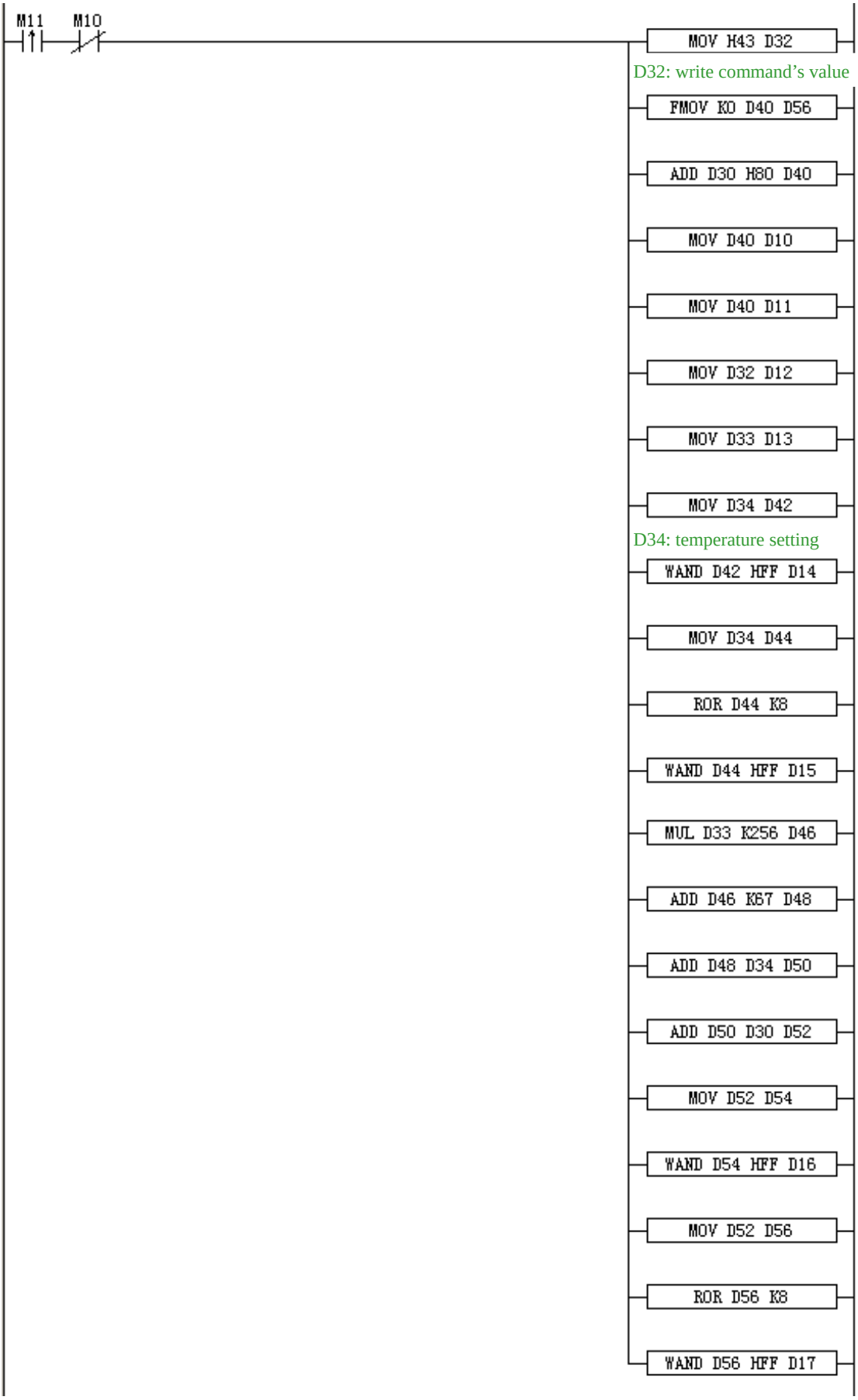
CRC check code: D36

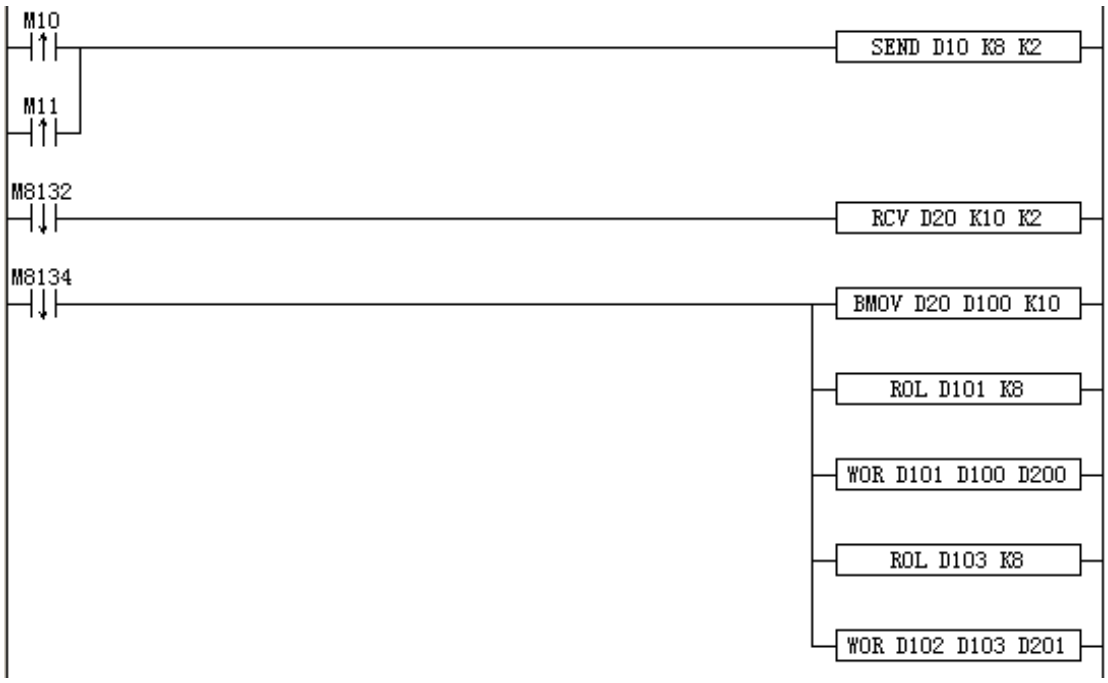
Temperature display: D200,D201

Format of sending data: 81H 81H 43H 00H c8H 00H 0cH 01H (display of the current temperature)
Setting of communication parameters: baud rate: 9600，8 bits data bit, 2 bits stop bit, no check.
Set FD8220=255；FD8221=5。
Note (both the host machine and the slave machine should use the version higher than V2.4)

Program：


M8002
MOV K1 D30
T200
T200 K4
T200
M10
M10
M11
MOV H52 D31
FMOV K0 D40 D56
ADD D30 H80 D40
MOV D40 D10
MOV D40 D11
MOV D31 D12
MOV D33 D13
MOV K0 D14
MOV K0 D15
MUL D33 K256 D42
ADD D42 K82 D44
ADD D44 D30 D52
MOV D52 D54
WAND D54 HFF D16
MOV D52 D56
ROR D56 K8
WAND D56 HFF D17


M11
M10
MOV H43 D32
D32: write command's value
FMOV K0 D40 D56
ADD D30 H80 D40
MOV D40 D10
MOV D40 D11
MOV D32 D12
MOV D33 D13
MOV D34 D42
D34: temperature setting
WAND D42 HFF D14
MOV D34 D44
ROR D44 K8
WAND D44 HFF D15
MUL D33 K256 D46
ADD D46 K67 D48
ADD D48 D34 D50
ADD D50 D30 D52
MOV D52 D54
WAND D54 HFF D16
MOV D52 D56
ROR D56 K8
WAND D56 HFF D17


M10
M11
M8132
M8134
SEND D10 K8 K2
RCV D20 K10 K2
BMOV D20 D100 K10
ROL D101 K8
WOR D101 D100 D200
ROL D103 K8
WOR D102 D103 D201

MEMO

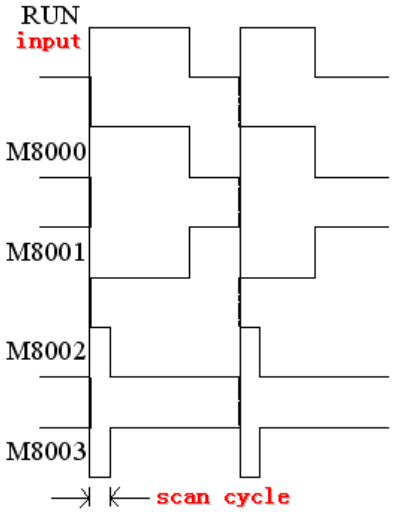
8. Appendix

This chapter gives some auxiliary information of XC series PLC.

8-1. List of special auxiliary relay、special data register

Special soft unit's type and its function

PC status（M）

ID	Function	Description	
M8000	Working normally ON coil		PLC be ON when running
M8001	Working normally OFF coil		PLC be OFF when running
M8002	Initial positive pulse coil		The first scan cycle is ON when PLC starts running
M8003	Initial negative pulse coil		The first scan cycle is OFF when PLC starts running
M8005	Battery voltage too low	Act when battery voltage abnormal too low	

PC status（D）

ID	Function	Description
D8002	Register's capacity	2…2K steps；4…4K steps；8…8K steps
D8005	Battery voltage	0.1V unit

Clock （M）

ID	Function	Description
M8010		
M8011	Shake with the cycle of 10ms	
M8012	Shake with the cycle of 100ms	
M8013	Shake with the cycle of 1	
M8014	Shake with the cycle of 1	
M8018	Bits of year	Defaulted is OFF（OFF：2；ON：4）

Flag （M）

ID	Function	Description
M8020	Zero	When plus/minus operation result is 0
M8021	Borrow	When borrow occurs in minus operation
M8022	Carry	When carry occurs in plus operation or overflow occurs in bit shift operation
M8023		
M8026	RAMP mode	
M8029		

Clock （D）

ID	Function	Description
D8010	The current scan cycle	Unit: 0.1ms
D8011	Mini value of scan time	Unit: 0.1ms
D8012	Max vale of scan time	Unit: 0.1ms
D8013	Second（clock）	0~59（BCD code format）
D8014	Minute（clock）	0~59（BCD code format）
D8015	Hour（clock）	0~23（BCD code format）
D8016	Date（clock）	0~31（BCD code format）
D8017	Month（clock）	0~12（BCD code format）
D8018	Year（clock）	2000~2099（BCD code format）
D8019	Week（clock）	0（Sunday）~6（Saturday）（BCD code format）

Flag（D）

ID	Function	Description
D8021	Model	Low byte
	Serial number	High byte
D8022	Compatible system’s version number	Low byte
	System’s version number	High byte
D8023	Compatible model’s version number	Low byte
	Model’s version number	High byte
D8024	Model’s information	Max 5 ASC and a “\0”
D8025		
D8026		
D8027	suitable host machine version	
D8028		
D8029		

PC mode（M）

ID	Function	Description
M8030	PLC initializing	
M8031	Non-retentive register clear	When driving this M, ON/OFF image memory of Y, M, S, TC and the current value of T, C, D are all cleared
M8032	Retentive register clear	
M8033	Register retentive stop	When PLC changes from RUN to STOP, leave all content in image register and data register
M8034	All output forbidden	Set PC's all external contacts to be OFF status
M8038	Parameter setting	Communication parameters set flag

PC mode（D）

ID	Function	Description
D8030		
D8031		
D8032		
D8033		
D8034		
D8035		
D8036		
D8037		
D8038		

Step ladder（M）

ID	Function	Description
M8041		
M8045	All output reset forbidden	When mode shifting, all output reset are forbidden
M8046	STL status act	When M8047 acts, act when any unit of S0~S999 turns to be ON

Interrupt（M）

ID	Function	Description
M8050 I00□	Forbid input interruption 0	After executing EI, even interruption allowed, but when M acts at this time, the correspond input interruption couldn't act separately E.g.： when M8050 is ON, interrupt I00□ is forbidden
M8051 I10□	Forbid input interruption 1	
M8052 I20□	Forbid input interruption 2	
M8053 I30□	Forbid input interruption 3	
M8054 I40□	Forbid input interruption 4	
M8055 I50□	Forbid input interruption 5	
M8056 I60□	Forbid time interruption 0	After executing EI, even interruption allowed, but when M acts at this time, the correspond input interruption couldn't act separately
M8057 I70□	Forbid time interruption 1	
M8058 I80□	Forbid time interruption 2	
M8059	Counter interrupt forbidden	Forbid interruption from I010~I060

Error check（M）

ID	Function	Description
M8067	Operation error	Power on and STOP->RUN check
M8070	Scan overtime	
M8071	No user program	Interior codes checking error
M8072	User program error	Execute code or collocate table check error

Error check（D）

ID	Function	Description
D8067	Execute error code’s ID	Error of divide
D8068	Lock occur error code’s ID	
D8069		
D8070	Scan time of overtime	Unit: 1ms
D8074	ID of Excursion register D	
D8097		
D8098		

Communication（M）

	ID	Function	Description
COM1	M8120		
	M8122	RS232 is sending flag	
	M8124	RS232 is receiving flag	
	M8125	Receive imperfect flag	Receiving finished normally, but the received data is less than the required
	M8127	Receive error flag	
	M8128	Receive correct flag	
	M8129	Timeout judgment flag	
COM2	M8130		
	M8132	RS232 is sending flag	
	M8134	RS232 is receiving flag	
	M8135	Receive imperfect flag	Receiving finished normally, but the received data is less than the required
	M8137	Receive error flag	
	M8138	Receive correct flag	
	M8139	Timeout judgment flag	
COM3	M8140		
	M8142	RS232 is sending flag	
	M8144	RS232 is receiving flag	
	M8145	Receive imperfect flag	Receiving finished normally, but the received data is less than the required
	M8147	Receive error flag	
	M8148	Receive correct flag	
	M8149	Timeout judgment flag	

Communication（D）

	ID	Function	Description
COM1	D8120		
	D8121		
	D8123	Data number received by RS232	
	D8126		
	D8127	Communication error code	7: hardware error 10: no start sign 8: CRC check error 11: no end sign 9: bureau ID error
	D8128		
	D8129		
COM2	D8130		
	D8131		
	D8133	Data number received by RS232	
	D8136		
	D8137	Communication error code	7: hardware error 10: no start sign 8: CRC check error 11: no end sign 9: bureau ID error
	D8138		
	D8139		
COM3	D8140		
	D8141		
	D8143	Data number received by RS232	
	D8146		
	D8147	Communication error code	7: hardware error 10: no start sign 8: CRC check error 11: no end sign 9: bureau ID error
	D8148		
	D8149		

High speed count（M）

ID	Counter ID	Function	Description
M8150	C600	Count finished sign	24 segments count finished, flag is 1
M8151	C602	Count finished sign	24 segments count finished, flag is 1
M8152	C604	Count finished sign	24 segments count finished, flag is 1
M8153	C606	Count finished sign	24 segments count finished, flag is 1
M8154	C608	Count finished sign	24 segments count finished, flag is 1
M8155	C610	Count finished sign	24 segments count finished, flag is 1
M8156	C612	Count finished sign	24 segments count finished, flag is 1
M8157	C614	Count finished sign	24 segments count finished, flag is 1
M8158	C616	Count finished sign	24 segments count finished, flag is 1
M8159	C618	Count finished sign	24 segments count finished, flag is 1
M8160	C620	Count finished sign	24 segments count finished, flag is 1
M8161	C622	Count finished sign	24 segments count finished, flag is 1
M8162	C624	Count finished sign	24 segments count finished, flag is 1
M8163	C626	Count finished sign	24 segments count finished, flag is 1
M8164	C628	Count finished sign	24 segments count finished, flag is 1
M8165	C630	Count finished sign	24 segments count finished, flag is 1
M8166	C632	Count finished sign	24 segments count finished, flag is 1
M8167	C634	Count finished sign	24 segments count finished, flag is 1
M8168	C636	Count finished sign	24 segments count finished, flag is 1
M8169	C638	Count finished sign	24 segments count finished, flag is 1

Pulse output（M）

ID	High frequency pulse ID	Function	Description
M8170	PULSE_1	Sending pulse flag	Be 1 at pulse sending
M8171		32 bits pulse sending overflow flag	Be 1 when overflow
M8172		Direction flag	1 is positive direction, the correspond direction port is ON
M8173	PULSE_2	Sending pulse flag	Be 1 at pulse sending
M8174		32 bits pulse sending overflow flag	Be 1 when overflow
M8175		Direction flag	1 is positive direction, the correspond direction port is ON
M8176	PULSE_3	Sending pulse flag	Be 1 at pulse sending
M8177		32 bits pulse sending overflow flag	Be 1 when overflow
M8178		Direction flag	1 is positive direction, the correspond direction port is ON
M8179	PULSE_4	Sending pulse flag	Be 1 at pulse sending
M8180		32 bits pulse sending overflow flag	Be 1 when overflow
M8181		Direction flag	1 is positive direction, the correspond direction port is ON
M8182	PULSE_5	Sending pulse flag	Be 1 at pulse sending
M8183		32 bits pulse sending overflow flag	Be 1 when overflow
M8184		Direction flag	1 is positive direction, the correspond direction port is ON

Positive/negative count

ID	Counter's ID	Function	Description
M8238	C300	Control of positive/negative count	0 is plus count, 1 is minus count, the defaulted is 0
……			

High speed count（D）

ID	Counter's ID	Function	Description
D8150	C600	The current segment (means No.n segment)	
D8151	C602	The current segment	
D8152	C604	The current segment	
D8153	C606	The current segment	
D8154	C608	The current segment	
D8155	C610	The current segment	
D8156	C612	The current segment	
D8157	C614	The current segment	
D8158	C616	The current segment	
D8159	C618	The current segment	
D8160	C620	The current segment	
D8161	C622	The current segment	
D8162	C624	The current segment	
D8163	C626	The current segment	
D8164	C628	The current segment	
D8165	C630	The current segment	
D8166	C632	The current segment	
D8167	C634	The current segment	
D8168	C636	The current segment	
D8169	C638	The current segment	

Pulse output（D）

Expansion's information（D）

ID	High frequency pulse ID	Function	Description
D8170	PULSE_1	The low 16 bits of accumulated pulse number	
D8171		The high 16 bits of accumulated pulse number	
D8172		The current segment (means No.n segment)	
D8173	PULSE_2	The low 16 bits of accumulated pulse number	
D8174		The high 16 bits of accumulated pulse number	
D8175		The current segment (means No.n segment)	
D8176	PULSE_3	The low 16 bits of accumulated pulse number	
D8177		The high 16 bits of accumulated pulse number	
D8178		The current segment (means No.n segment)	
D8179	PULSE_4	The low 16 bits of accumulated pulse number	
D8180		The high 16 bits of accumulated pulse number	
D8181		The current segment (means No.n segment)	
D8182	PULSE_5	The low 16 bits of accumulated pulse number	
D8183		The high 16 bits of accumulated pulse number	
D8184		The current segment (means No.n segment)	
D8190	PULSE_1	The low 16 bits of accumulated pulse number	
D8191		The high 16 bits of accumulated pulse number	
D8192	PULSE_2	The low 16 bits of accumulated pulse number	
D8193		The high 16 bits of accumulated pulse number	
D8194	PULSE_3	The low 16 bits of accumulated pulse number	
D8195		The high 16 bits of accumulated pulse number	
D8196	PULSE_4	The low 16 bits of accumulated pulse number	
D8197		The high 16 bits of accumulated pulse number	
D8198	PULSE_5	The low 16 bits of accumulated pulse number	
D8199		The high 16 bits of accumulated pulse number	

Unit	Type	ID（as register）	Max I/O//channels
Expansion 1#	Input switch quantity X	X100~X137	32 points
	Output switch quantity Y	Y100~Y137	32 points
	Input analog ID	ID100~ID131	16 channels
	Output analog QD	QD100~QD131	16 channels
	Module’s set value D	D8250~D8259	-
Expansion 2#	Input switch quantity X	X200~X237	32 points
	Output switch quantity Y	Y200~Y237	32 points
	Input analog ID	ID200~ID231	16 channels
	Output analog QD	QD200~QD231	16 channels
	Module’s set value D	D8260~D8269	-
Expansion 3#	Input switch quantity X	X300~X337	32 points
	Output switch quantity Y	Y300~Y337	32 points
	Input analog ID	ID300~ID331	16 channels
	Output analog QD	QD300~QD331	16 channels
	Module’s set value D	D8270~D8279	-
Expansion 4#	Input switch quantity X	X400~X437	32 points
	Output switch quantity Y	Y400~Y437	32 points
	Input analog ID	ID400~ID431	16 channels
	Output analog QD	QD400~QD431	16 channels
	Module’s set value D	D8280~D8289	-
Expansion 5#	Input switch quantity X	X500~X537	32 points
	Output switch quantity Y	Y500~Y537	32 points
	Input analog ID	ID500~ID531	16 channels
	Output analog QD	QD500~QD531	16 channels
	Module’s set value D	D8290~D8299	-
Expansion 6#	Input switch quantity X	X600~X637	32 points
	Output switch quantity Y	Y600~Y637	32 points
	Input analog ID	ID600~ID631	16 channels
	Output analog QD	QD600~QD631	16 channels
	Module’s set value D	D8300~D8309	-
Expansion 7#	Input switch quantity X	X700~X737	32 points
	Output switch quantity Y	Y700~Y737	32 points
	Input analog ID	ID700~ID731	16 channels
	Output analog QD	QD700~QD731	16 channels
	Module’s set value D	D8310~D8319	-
BD Expansion	Input switch quantity X	X1000~X1037	32 points
	Output switch quantity Y	Y1000~Y1037	32 points
	Input analog ID	ID1000~ID1031	16 channels
	Output analog QD	QD1000~QD1031	16 channels
	Module’s set value D	D8320~D8329	-

8-2. List of special FLASH data register SFD

1、 I filter

Number	Function	Description
FD8000	X0~X17 input filter time value	
FD8002	X20~X37 input filter time value	
FD8003	X40~X57 input filter time value	
FD8004		
FD8005		
FD8006		
FD8007		
FD8008		
FD8009		

2、 I mapping

Number	Function	Description
FD8010	X00 corresponds with I**	X0 corresponds with the number of input image I**
FD8011	X01 corresponds with I**	
FD8012	X02 corresponds with I**	
……	……	
FD8073	X77 corresponds with I**	

3、 O mapped

Number	Function	Description
FD8074	Y00 corresponds with I**	Y0 corresponds with the number of input image O**
FD8075	Y01 corresponds with I**	
FD8076	Y02 corresponds with I**	
……	……	
FD8137	Y77 corresponds with I**	

4、 I property

Number	Function	Description
FD8138	X00 property	0: positive logic； others: negative logic
FD8139	X01 property	
FD8140	X02 property	
……	……	
FD8201	X77 property	

5、 Device's power failure retentive area

Number	Function	Description
FD8202	Start tag of D power failure	

	store area	
FD8203	Start tag of M power failure store area	
FD8204	Start tag of T power failure store area	
FD8205	Start tag of C power failure store area	
FD8206	Start tag of S power failure store area	

6、 Communication

COM1	Number	Function	Description
	FD8210	Communicate mode	255 is free format， 1~254 bits modbus station ID
	FD8211	Communicate format	Baud rate，data bit，stop bit，checkout
	FD8212	Judgment time of ASC timeout	Unit: ms
	FD8213	Judgment time of reply timeout	Unit: ms，if set to be 0, it means no timeout waiting
	FD8214	Start ASC	High 8 bits be of no effect
	FD8215	End ASC	Low 8 bits be of no effect
	FD8216	Free format setting	8/16 bits cushion, have/no start bit, have/no end bit,
COM2	FD8220	Communicate mode	255 is free format， 1~254 bits modbus station ID
	FD8221	Communicate format	Baud rate，data bit，stop bit，checkout
	FD8222	Judgment time of ASC timeout	High 8 bits be of no effect
	FD8223	Judgment time of reply timeout	Low 8 bits be of no effect
	FD8224	Start ASC	Unit: ms
	FD8225	End ASC	Unit: ms，if set to be 0, it means no timeout waiting
	FD8226	Free format setting	8/16 bits cushion, have/no start bit, have/no end bit
COM3	FD8230	Communicate mode	255 is free format， 1~254 bits modbus station ID
	FD8231	Communicate format	Baud rate，data bit，stop bit，checkout
	FD8232	Judgment time of ASC timeout	High 8 bits be of no effect

	FD8233	Judgment time of reply timeout	Low 8 bits be of no effect
	FD8234	Start ASC	Unit: ms
	FD8235	End ASC	Unit: ms，if set to be 0, it means no timeout waiting
	FD8236	Free format setting	8/16 bits cushion, have/no start bit, have/no end bit

Remark

Some of instructions stated in this manual are still in developing, please note!

1、Applied instructions

Data Shift	SFTL	Bit shift left
	SFTR	Bit shift right
	WSFL	Word shift left
	WSFR	Word shift right
Data convert	FLT	16 bits integer converts to be floating
	FLTD	64 bits integer converts to be floating
	INT	Floating converts to be integer
	BIN	BCD converts to be binary
	BCD	Binary converts to be BCD
Clock operation	TCMP	Clock data compare
	TZCP	Zone compare of clock data
	TADD	Addition of clock data
	TSUB	Subtraction of clock data
Floating operation	ECMP	Floating compare
	EZCP	Zone compare of floating
	EADD	Addition of floating
	ESUB	Subtracting of floating
	EMUL	Multiplication of floating
	EDIV	Division of floating
	ESQR	Extraction of floating
	SIN	SIN operation of floating
	COS	COS operation of floating
	TAN	TAN operation of floating

2、special functions

1） Alterable frequency pulse output [PLSF]

2） Frequency testing [FRQM]